

DEEP LEARNING MODELS FOR MODELING CELLULAR TRANSCRIPTION SYSTEMS

by

Lujia Chen

BS, Department of Biotechnology, University of Science and Technology Beijing, China, 2009

MS, Department of Biomedical Informatics, University of Pittsburgh, USA, 2012

Submitted to the Graduate Faculty of
School of Medicine in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

SCHOOL OF MEDICINE

This thesis was presented

by

Lujia Chen

It was defended on

August 05, 2016

and evaluated by

Greg Cooper, MD, PHD, Department of Biomedical Informatics, University of Pittsburgh

Shyam Visweswaran, MD, PHD, Department of Biomedical Informatics, University of
Pittsburgh

Nathan Clark, PHD, Department of Computational and Systems Biology, University of
Pittsburgh

Thesis Director/Dissertation Advisor: Xinghua Lu, MD, PHD, Department of Biomedical
Informatics, University of Pittsburgh

Copyright © by Lujia Chen

2016

DEEP LEARNING MODELS FOR MODELING CELLULAR TRANSCRIPTION SYSTEMS

Lujia Chen, M.S.

University of Pittsburgh, 2016

Cellular signal transduction system (CSTS) plays a fundamental role in maintaining homeostasis of a cell by detecting changes in its environment and orchestrates response. Perturbations of CSTS lead to diseases such as cancers. Almost all CSTSs are involved in regulating the expression of certain genes and leading to signature changes in gene expression. Therefore, the gene expression profile of a cell is the readout of the state of its CSTS and could be used to infer CSTS. However, a gene expression profile is a convoluted mixture of the responses to all active signaling pathways in cells. Therefore it is difficult to find the genes associated with an individual pathway. An efficient way of de-convoluting signals embedded in the gene expression profile is needed.

At the beginning of the thesis, we applied Pearson correlation coefficient analysis to study cellular signals transduced from ceramide species (lipids) to genes. We found significant correlations between specific ceramide species or ceramide groups and gene expression. We showed that various dihydroceramide families regulated distinct subsets of target genes predicted to participate in distinct biologic processes. However, it's well known that the signaling pathway structure is hierarchical. Useful information may not be fully detected if only linear models are used to study CSTS. More complex non-linear models are needed to represent the hierarchical structure of CSTS. This motivated us to investigate contemporary deep learning models (DLMs).

Later, we applied various deep hierarchical models to learn a distributed representation of statistical structures embedded in transcriptomic data. The models learn and represent the

hierarchical organization of transcriptomic machinery. Besides, they provide an abstract representation of the statistical structure of transcriptomic data with flexibility and different degrees of granularity. We showed that deep hierarchical models were capable of learning biologically sensible representations of the data (e.g., the hidden units in the first hidden layer could represent transcription factors) and revealing novel insights regarding the machinery regulating gene expression. We also showed that the model outperformed state-of-the-art methods such as Elastic-Net Linear Regression, Support Vector Machine and Non-Negative Matrix Factorization.

TABLE OF CONTENTS

PREFACE.....	XVI
1.0 OVERALL HYPOTHESIS.....	1
2.0 OVERALL INTRODUCTION.....	2
2.1 SIGNALING TRANSDUCTION	2
2.1.1 Understanding signal transduction is fundamental in cell biology	2
2.1.2 The relationship between signaling transduction, disease, therapy and drug discovery	4
2.1.3 Yeast signaling is a good model system for studying signaling pathways...	5
2.1.4 System perturbation is a powerful tool for studying signaling pathway	6
2.1.5 Using high-throughput transcriptomic data as signaling readout	7
2.1.6 Feature selection, dimension reduction and pattern recognition	8
2.1.7 Using latent variable models to represent signals embedded in transcriptomic data.....	10
2.1.7.1 Cluster Analysis (Hierarchical clustering analysis)	11
2.1.7.2 Principle component analysis (PCA)	12
2.1.7.3 Non-negative matrix factorization (NMF)	15
2.1.8 Discovering signals embedded in transcriptomic data by functional analysis of gene sets.....	17

2.1.8.1	Gene Ontology (GO) analysis	18
2.1.8.2	Network-extracted ontology (NeXO).....	18
2.1.8.3	Kyoto Encyclopedia of Genes and Genomes (KEGG)	19
2.1.8.4	Gene Set Enrichment Analysis (GSEA)	19
2.2	THE NEEDS FOR NOVEL MODELS FOR STUDYING THE TRANSCRIPTOMIC REGULATION SYSTEM.....	20
2.3	DEEP LEARNING	21
2.3.1	History of deep learning	21
2.3.2	Basic structure of artificial neural networks.....	22
2.3.2.1	Activation function at each neuron	22
2.3.2.2	Backpropagation to train a multi-layer network	26
2.3.3	Supervised and unsupervised learning	27
2.3.4	Deep learning models.....	29
2.3.4.1	Deep belief network (DBN)	29
2.3.4.2	Sparse Deep Belief Network	31
2.3.4.3	Dropout Neural Network.....	31
2.3.4.4	Deep Boltzmann Machine (DBM)	33
2.3.4.5	Multimodal Deep Boltzmann Machine	34
2.3.4.6	Convolutional Neural Network (CNN)	35
2.3.5	Identification and visualization of information represented by latent variables in hierarchical models	36

2.3.6	Application of DLMs in different machine learning domains	41
2.3.7	Application of DLMs in biomedical fields	44
2.3.8	The potential of DLMs for drug sensitivity prediction, cancer subtype classification and personalized medicine	48
3.0	INTRODUCTION OF STUDIES	50
4.0	CHAPTER 1: DISTINCT SIGNALING OF CERAMIDE SPECIES IN YEAST REVEALED THROUGH SYSTEMATIC PERTURBATION AND SYSTEMS BIOLOGY ANALYSES	52
4.1	INTRODUCTION.....	52
4.2	MATERIAL AND METHODS	54
4.3	RESULTS	59
4.4	DISCUSSION	73
5.0	CHAPTER 2: TRANS-SPECIES LEARNING OF CELLULAR SIGNALING SYSTEMS WITH BIMODAL DEEP BELIEF NETWORKS	76
5.1	INTRODUCTION.....	76
5.2	METHODS	80
5.2.1	Restricted Boltzmann Machine (RBMs)	81
5.2.2	Learning Parameter of RBM model.....	82
5.2.3	Learning a Deep Belief Network.....	83
5.2.4	Bimodal DBN.....	84
5.2.5	Semi-Restricted Bimodal Deep Belief Network (sbDBN).....	87
5.2.6	Performance evaluation.....	88
5.2.7	Model Selection.....	89

5.2.8	Baseline predictive models	90
5.3	RESULTS	90
5.3.1	The Data.....	90
5.3.2	Model selection results.....	91
5.3.3	Comparison among different models	92
5.3.4	Biological interpretation of learned edges between proteins in sbDBN....	93
5.4	DISCUSSION	95
6.0	CHAPTER 3: LEARNING A HIERARCHICAL REPRESENTATION OF THE YEAST TRANSCRIPTOMIC MACHINERY USING AN DEEP AUTOENCODER MODEL 98	
6.1	INTRODUCTION.....	98
6.2	METHODS	101
6.2.1	Restricted Boltzmann Machines (RBMs).....	102
6.2.2	Autoencoder.....	103
6.2.3	Sparse autoencoder	103
6.2.4	Non-negative matrix factorization.....	105
6.2.5	Model selection of autoencoder and sparse autoencoder	105
6.2.6	Mapping between the hidden units and known biological components..	107
6.2.7	Consensus clustering of experiment samples.....	108
6.2.8	Finding pheromone related hidden units.....	108
6.2.9	Gene ontology analysis.....	109
6.2.10	Incorporation of Prior Knowledge into the DBN model	110

6.2.11	Identification and visualization of information represented by latent variables in higher hidden layers.....	112
6.3	RESULTS AND DISCUSSION	112
6.3.1	Training different models for representing yeast transcriptomic machinery	112
6.3.2	Distributed representation enhances discovery of signals of TFs.....	118
6.3.3	Latent variables can capture the information of signaling pathways	121
6.3.4	The hierarchical structure captures signals of different degrees of abstraction	122
6.3.5	Concise representation enhances the discovery of global patterns	126
6.3.6	Information embedded in data is consistently represented in different hidden layers.....	128
6.3.7	Incorporaing prior domain knowledge into DBN model	131
6.4	CONCLUSION	132
7.0	OVERALL CONCLUSION.....	133
8.0	OVERALL DISCUSSION	134
9.0	FUTURE WORK	136
9.1.1	Using DLMs to perform translational studies of human cancer	136
9.1.2	Using multimodal DBNs to incorporate more data types to study signaling networks.....	136
	APPENDIX A	138
	APPENDIX B	151
	APPENDIX C	157

BIBLIOGRAPHY	160
---------------------------	------------

LIST OF FIGURES

Figure 2.1. Mating-pheromone response pathway in budding yeast.	3
Figure 2.2. Illustration of the difference between the number of copies of the HER2 gene in normal cells and HER2 overexpressed cancer cells.	5
Figure 2.3. Illustration of principle component analysis (PCA).	14
Figure 2.4. Basic structure of an artificial neural network.	23
Figure 2.5. Illustration of the activation function of a neural network.	23
Figure 2.6. Sigmoid activation function.	24
Figure 2.7. Hyperbolic tangent activation function.	25
Figure 2.8. Rectified linear unit (ReLU) activation function.	26
Figure 2.9. Illustration of backpropagation.	27
Figure 2.10 Supervised learning and unsupervised learning.	28
Figure 2.11. Autoencoder.	30
Figure 2.12. Restricted Boltzmann Machine (RBM).	30
Figure 2.13. Dropout Deep Belief Network.	32
Figure 2.14. Presence of a unit at training time and test time.	33
Figure 2.15 A three-layer Deep Boltzmann Machine (DBM) and a three-layer Deep Belief Network (DBN).	34
Figure 2.16. Multimodal Deep Boltzmann Machine (DBM).	35

Figure 2.17. Convolutional Neural Network (CNN).	36
Figure 2.18. Visualization of the first hidden layer by performing weight normalization.	38
Figure 2.19. Comparison of three visualization methods.	41
Figure 2.20. Application of DLMs in computer vision.	42
Figure 2.21. Similar information embedded in image and text.	44
Figure 2.22. Number of deep learning papers in all fields vs. bioinformatics field in the past decade.	45
Figure 4.1. Overall strategy of the study.....	61
Figure 4.2. Lipidomic analysis.....	62
Figure 4.3. Assessing the correlation between lipid abundance and gene expression.....	66
Figure 4.4. Modeling relationship between lipidomic and gene expression data.	69
Figure 5.1. Trans-species learning task specification.	77
Figure 5.2. Graph representation of the Deep Belief Network and related models.....	81
Figure 5.3. Training DBN models.	85
Figure 5.4. Prediction with bDBN and sbDBN models.....	86
Figure 5.5. ROC and RPC curves of different models.	93
Figure 5.6. Protein correlation network learned from the 4-layered sbDBN.....	94
Figure 6.1. Overview of studying molecular signaling transduction using an autoencoder.....	100
Figure 6.2. Pheromone signaling pathway related proteins.....	109
Figure 6.3. Histogram of the expected states of hidden units (probability of hidden units to be on) in the first hidden layer for the conventional autoencoder (A) and sparse autoencoder (B) respectively.	116

Figure 6.4. Mapping between transcription factors (TFs) and hidden variables in the first hidden layer.....	119
Figure 6.5. Example of hierarchical Gene Ontology (GO) map for hierarchical hidden structure.	124
Figure 6.6. Example of multiple GO terms associated with a hidden unit in a higher hidden layer.	125
Figure 6.7. Example of the representation of a hidden unit related to pheromone signaling pathway.	126
Figure 6.8. Clustering of experiment samples represented using original gene expression data (A), NMF metagenes (B) and expected state of hidden nodes in the first hidden layer (C).....	128
Figure 6.9. Cluster by cluster clustering for clusters in the 1 st hidden layer and the 2 nd hidden layer.....	130
Figure 9.1. Two-layered structure of RBM.	140
Figure 9.2. Illustration of Gibbs sampling.	143
Figure 9.3. Comparison of reconstruction errors corresponding to DBN models with different settings.	154
Figure 9.4. <i>S. cerevisiae</i> sphingolipid metabolism.	157
Figure 9.5. Role of Ydc1 in mediating the impact of heat stress on gene expression.	158
Figure 9.6. Experimental validation of lipid-dependent phenotypes.	159

LIST OF TABLES

Table 1. Proteins and stimuli involved in this study	91
Table 2. Leave-one-out accuracy scores of models	93
Table 3. Binary values of pheromone-related samples and states of hidden units	110
Table 4. Contingency table between the states of samples and the states of each hidden unit...	110
Table 5. Reconstruction error of models with different architectures	116
Table 6. BIC scores of different models	117
Table 7. Quantitative comparisons between the four-layered sparse DBN and non-hierarchical NMF.....	121
Table 8. Two hidden units found to be the most related to pheromone signaling.....	124
Table 9. Reconstruction error of DBN with and without prior knowledge	131
Table 10. Generated states of hidden units in the hidden layer j	152
Table 11. Comparison of reconstruction errors among DBN models with different settings	155
Table 12. Quantitative comparison among DBN models with different settings	156

PREFACE

When looking back on my graduate study and life in the past years, I feel grateful to many wonderful people. I would first like to thank my advisor, Dr. Xinghua Lu, for providing exceptional support and advice throughout my PhD study. He always motivated and encouraged me to push forward during my dissertation research. He was always there whenever I needed suggestions on problems in my experiments. I would also like to thank the members of my Advisory Committee - Drs. Greg Cooper, Shyam Visweswaran and Nathan Clark - for monitoring my progress and providing prudent advice.

I would also like to thank all the members of Dr. Lu's laboratory: Vicky Chen; Chunhui Cai; Jonathan Young; Michael Ding and Xueer Chen. Their feedback and openness has made studying here a much more rewarding experience. In particular, I would like to thank Vicky and Chunhui for their support both professionally and personally. I would also like to thank Toni Porterfield for all of her help and support during my studies here at DBMI. Without her help scheduling and coordinating all of the meetings, defenses, and milestones along the way, my time here would have been much more difficult. I'm so grateful to be a member of the DBMI family.

Finally, I would like to dedicate my dissertation to my parents, husband and daughter for their continual and unwavering support. I thank you all for supporting me throughout this wonderful journey in Pittsburgh.

1.0 OVERALL HYPOTHESIS

This dissertation studied the potential of using deep hierarchical models, including the deep belief network (DBN) and its derived models, to simulate the hierarchical cellular signaling system. We hypothesize that deep learning models (DLMs) could learn a distributed representation of statistical structures embedded in the transcriptomic data, and are capable of learning biologically sensible representations of the data and revealing novel insights regarding the machinery regulating gene expression. Besides this, we also hypothesize that DLMs (e.g., a bimodal deep belief network) could represent the common hierarchical signal-encoding mechanism between different species (e.g., human and rat) to perform trans-species learning.

Benefitting from the hierarchically organized latent variables capable of capturing the statistical structures in the observed data (e.g., transcriptomic data and proteomic data) in a distributed fashion, DLMs are suited for performing both the reconstruction task and the classification task. We hypothesize that DLMs could outperform state-of-the-art non-hierarchical models such as generalized linear model (Pearson correlation coefficient analysis and Elastic-net regression), principle component analysis (PCA), non-negative matrix factorization (NMF), and support vector machine (SVM).

2.0 OVERALL INTRODUCTION

2.1 SIGNALING TRANSDUCTION

2.1.1 Understanding signal transduction is fundamental in cell biology

Signal transduction is a fundamental process of living cells that controls cellular fates such as growth, proliferation, and survival. Signal transduction occurs when an extracellular or intracellular signaling molecule activates a specific receptor located on the cell surface or inside the cell. In turn, this receptor triggers a biochemical chain of events inside the cell creating a response and eventually alters certain cellular process, such as entry into a mitotic state. To be concise, signal transduction involves the intracellular signaling initiated by intracellular or extracellular signals such as hormones, growth factors, neurotransmitters, and cytokines. It is followed by signals transduced to downstream transcription factors that ultimately alter gene expression (Lee and McCubrey 2002).

As a concrete example, one of the well-known signaling transduction pathways in yeast is pheromone signaling transduction. Pheromones are chemicals capable of acting outside the body of the secreting individual to impact the behavior of the receiving individual (Gruhler, Olsen et al. 2005). Sex pheromones are involved in mating signaling whose processes are: 1) the pheromone first binds to its G-protein coupled receptor that leads to $G\alpha$ activation and

dissociation from the $G\beta\gamma$ heterodimer; 2) a conserved MAPK cascade is activated that leads to the transcription of mating-specific genes, cell polarization in the direction of partner cells and subsequent fusion of mating pairs (Figure 2.1).

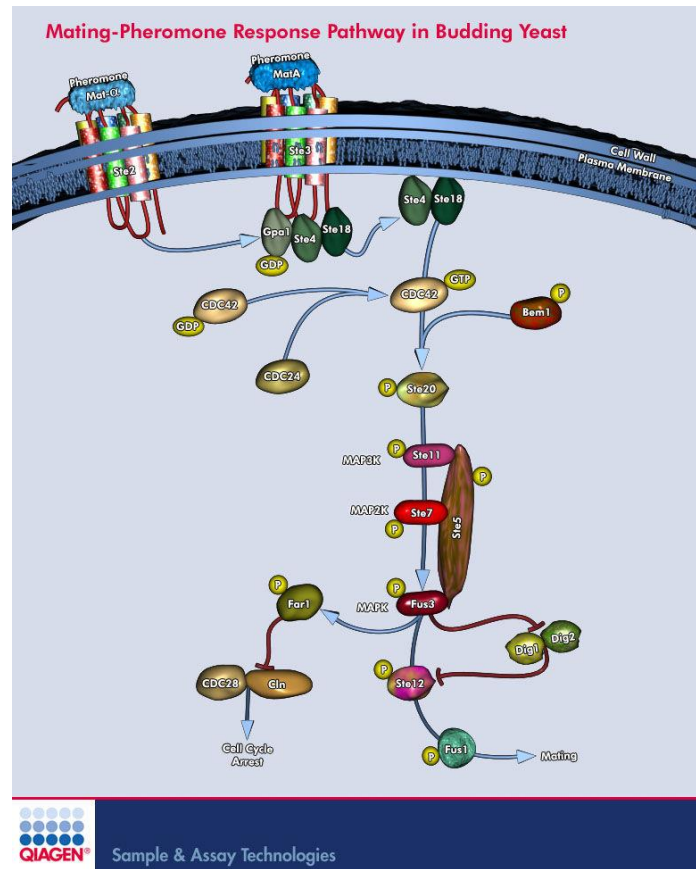


Figure 2.1. Mating-pheromone response pathway in budding yeast.(QIAGEN)

2.1.2 The relationship between signaling transduction, disease, therapy and drug discovery

Aberrations in cellular signaling systems will affect the capability of a cell to regulate homeostasis and its responses to environmental changes. In humans, when the function of proteins in signaling cascades are perturbed (e.g., due to mutated genes or copy number variations), transduction of signals will be disturbed which could lead to disease. Let's use the HER2 signaling pathway as an example. HER2 is a human receptor protein for epidermal growth factor, and it is one of the well-studied breast cancer signaling pathways (Slamon, Leyland-Jones et al. 2001). If the HER2 gene is amplified, the extra copies of the HER2 gene (Yarden 2001) (Figure 2.2) often result in over-expression of the HER2 protein, leading to activation of the protein and its downstream signals. The perturbed HER2 signaling pathway then leads to the uncontrolled growth and survival of cancer tumors.

When researchers realized the influence of perturbed signaling pathways on disease, drugs targeting specific signaling pathways were designed. When the signaling transductions are well studied in disease, such as HER2 signaling in breast cancer, we could use them as therapeutic targets. We treat patients with HER2 targeted drugs that could block the aberrant signals resulting from HER2 amplification (Goldhirsch, Ingle et al. 2009). Another example is sphingolipid-1-phosphate (S1P). It is verified that S1P and ceramide have profound effects on Glioblastoma multiforme (GBM) cells that are involved in brain tumors, with ceramide causing cell death and S1P leading to cell survival, proliferation and invasion. This kind of brain tumor could be treated with therapy targeting S1P and ceramide related pathways (Van Brocklyn 2007). Due to the critical importance of signaling transduction in biological process and disease study,

this dissertation applies new statistical and computational methods to better study signaling transduction.

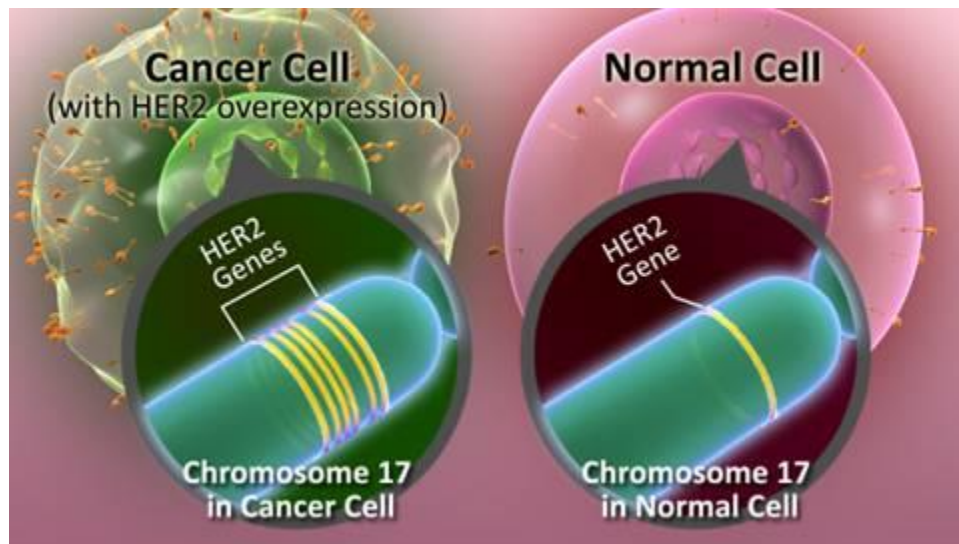


Figure 2.2. Illustration of the difference between the number of copies of the HER2 gene in normal cells and HER2 overexpressed cancer cells. (Shaffer 2011)

2.1.3 Yeast signaling is a good model system for studying signaling pathways

In this thesis, I applied a novel model to study signaling transduction in yeast. The complexity of signaling transduction increases when moving from unicellular organisms to multicellular organisms. *Saccharomyces cerevisiae* (the baker's yeast) is a eukaryotic model organism for studying genetics, molecular biology, and cell biology. An important body of contemporary knowledge of the components, interrelationships, and regulators involved in signaling transductions was first discovered in yeast (Dohlman and Slessareva 2006). The study of yeast

provides important insight into the understanding of signaling networks such as Ras signaling (Tamanoi 2011). For a new model (deep learning models in this proposal) applied to study signaling networks, unicellular signal transduction is a good start, because the genetic interactions in these organisms such as yeast are relatively less complex. What's more, it is much easier to validate the newfound pathways in yeast than in multicellular organisms, such as rats and humans. If we successfully validate the feasibility of the new model in yeast, we could then apply the model to study more complex systems, such as human cancer, to find the specific signaling pathways involved in different types of cancer.

2.1.4 System perturbation is a powerful tool for studying signaling pathway

Another reason we choose yeast is that it is easy to genetically manipulate yeast cells, rendering yeast cells amenable to systematic perturbations. For example, there is a library of yeast strains with distinct genetic manipulations (Tong, Evangelista et al. 2001, Giaever, Chu et al. 2002, Boone, Bussey et al. 2007, Li, Vizeacoumar et al. 2011). This systematic analysis allows us to study the genetic interactions and networks (Sturtevant 1956). The concept of genetic interaction, how genes work interactively in a system instead of individually to lead to one phenotype, was profoundly influenced by these studies. The functions of genes in *Saccharomyces cerevisiae* could be reflected by its influence on cellular functions (Tong, Evangelista et al. 2001).

Many recent studies use yeast to study the genetic interactions and networks in which powerful functional genomic tools allow systematic analyses to be performed (Hughes, Robinson et al. 2004, Dolinski and Botstein 2005). The functional genomic tools include deletion-mutant collections and essential gene mutant collections (Boone, Bussey et al. 2007). Gene expression under thousands of experimental conditions, such as heat stress and mutation, were collected.

With systematic perturbation data, the genetic interaction and the causality relationship between genes can be inferred (Montefusco, Chen et al. 2013).

2.1.5 Using high-throughput transcriptomic data as signaling readout

In yeast genetics, the knowledge of gene function is often achieved by identifying the alteration of a gene (as a result of natural variance or experimental manipulation), followed by studying the impact of the alteration to certain phenotypes, such as cell growth and certain markers. Since changes in protein function often eventually lead to gene expression changes, gene expression data can be thought of as molecular phenotypes of genetic alterations (Brand 1993, Hughes, Marton et al. 2000).

With the availability of an increasingly large volume of comprehensive gene expression data collected under a large number of perturbation conditions, we can use statistical and computational methods to study genome-wide patterns of gene expression (genes that are co-expressed and perform related functions) by capturing their expression covariance structures under different experimental perturbations. We can further learn the functional relationship of perturbed genes by checking whether they share similar gene expression patterns (Eisen, Spellman et al. 1998).

Using gene expression as molecular phenotypes creates challenges. 1) The gene expression profile of a collection of cells is a convoluted mixture of the outcome of all the signals in the cell that regulate gene expression. Thus it is difficult to determine which gene is regulated by which signal. 2) Gene expression is high-dimensional data, and it is difficult to interpret data at the individual gene level. Researchers have developed different methods to capture the major coordinated changes in gene expression data as a means to represent biological

signals that regulate gene expression. 3) Compared with the large number of genes, the number of experiment samples collected is relatively small. From a statistical view, this means that the number of variables/features is large and the number of samples is small. This easily increases the difficulty of prediction and leads to the over-fitting problem. Therefore, efficient ways of selecting features and reducing dimension are needed.

2.1.6 Feature selection, dimension reduction and pattern recognition

The number of genes measured by high throughput technology is always up to several tens of thousands of genes depending on the specific species (Hertzberg and Pope 2000). Transcriptomic data (measured by microarray and next-generation sequencing technologies) poses a great challenge for computational techniques, because of much redundant information embedded in the high-dimensional gene expression dataset. The redundancy is caused by the fact that there are many highly correlated genes. The truly independent genes are much fewer than the number of genes in the original dataset. Therefore, it's better at recognizing genes with similar expression patterns and studying them together instead of individually. Besides, compared to the large number of genes involved, available training datasets generally have a fairly small sample size (Antoniadis, Lambert-Lacroix et al. 2003). For classification task, this easily leads to the problem of over-fitting. The fact listed above increases the difficulty of studying biological processes, particularly at the genomics level (Saeys, Inza et al. 2007).

To reduce the redundant information and unwanted noisy embedded in gene expression data, machine learning methods, such as feature selection and dimension reduction, are often applied. In the study of transcriptomic data, each gene is regarded as a feature. Feature selection is the process of selecting a small subset of features from the original relatively large set of

features (Ding and Peng 2005). It is one of the methods whose main purpose is to get rid of redundant features without incurring much loss of information. In the meantime, it shortens training times and enhances generalization by reducing over-fitting. In the context of classification, feature selection techniques can be organized into three categories depending on how they combine feature selection with the construction of the classification model: filter methods, wrapper methods, and embedded methods (Saeys, Inza et al. 2007). For filter methods, they analyze intrinsic properties of the data and do not incorporate learning algorithms. Each feature is assigned a score based on a the statistical measurement such as information gain, correlation coefficient scores and Relief-F (Wang 2004). Features are ranked and selected according to the scores. For wrapper methods, they use a learning algorithm to measure the quality of the subsets of features without incorporating the structure of classification. Subsets of features are searched, evaluated and selected by applying search processes such as a best-first search (Kohavi 1997). For embedded methods, they combine learning with feature selection. The most common type of embedded methods are regularization methods such as lasso regression (Tibshirani 1996), ridge regression (Tibshirani 1996) and elastic net regression (Zou 2005). Features best contributing to accuracy are learned. Dimension reduction is the process of applying a transformation on the feature vector that results in a new numeric vector with fewer features (Antoniadis, Lambert-Lacroix et al. 2003). For dimension reduction, there are methods such as principle component analysis (PCA) (2.1.7.2) (Jolliffe 2002) and non-negative matrix factorization (NMF) (2.1.7.3) (Devarajan 2008). There are also many existing software and libraries for feature selection such as WEKA (Hall), R package “Caret” (Kursa 2010) and Feature Selection Toolbox (FST) (Somol 2001).

One of the limitations of feature selection is that the relatively smaller dataset is selected based on their discriminative power on the target class. They are not maximally representative of the original space covered by the entire dataset. The feature set may represent one or several dominant characteristics of the targeted phenotypes, but there could still be narrow regions of the relevant space that are not covered. Thus the generalizability of the feature set is limited. However, deep learning methods (e.g., deep belief network) mentioned later in my dissertation would avoid this problem by reconstructing features instead of selecting features. The deep learning models construct low-dimensional features to recover as much information embedded in the original high-dimensional features as possible.

2.1.7 Using latent variable models to represent signals embedded in transcriptomic data

Using gene expression as molecular phenotypes to discover signals embedded in the transcriptomic data is confronted with multiple challenges. 1) The gene expression profile of a collection of cells is a convoluted mixture of the outcome of all the signals in the cells that regulate gene expression. Thus it is difficult to determine which gene is regulated by which signal. Often, this challenge can be partially addressed by synchronizing cells using physiological or pathological conditions. For example, (Spellman, Sherlock et al. 1998) used yeast pheromone to synchronize cells in a culture during a yeast cell cycle study. As another example, the expression signature of a perturbed pathway may be consistently expressed in the majority of tumor cells hosting the genomic alterations. 2) The gene expression profile of a cell at a given time is a convolution of the outcomes of all cellular signals that are actively regulating transcription at the time of sample collection. As such, it is a challenge to de-convolute the signals and identify transcriptomic changes regulated by an individual pathway. 3) Compared

with the number of genes involved in transcriptomic data, the number of experiment samples collected is usually relatively small. Much of the effort of discovering signals embedded in transcriptomic data concentrates on discovering statistical structures that capture the relationships among co-expressed genes as an indication that these genes are regulated by a common signaling pathway. This section reviews different statistical methods that are commonly used to identify patterns of transcriptomic data.

2.1.7.1 Cluster Analysis (Hierarchical clustering analysis)

Clustering is a method of finding the similarities and dissimilarities of genes. It belongs to unsupervised learning without pre-defined classes. It is often one of the first steps in gene expression analysis (Eisen, Spellman et al. 1998). Among different clustering algorithms (Bendor, Shamir et al. 1999), hierarchical clustering is one of the most frequently used methods of clustering genes based on their expression data under different experimental conditions (Cameron, Middleton et al. 2012). Hierarchical clustering progressively groups genes exhibiting similar expression profiles across multiple conditions, and the result is a multi-level hierarchy tree where clusters at one layer are connected to clusters at the next higher layer. Different metrics may be used by hierarchical clustering to measure similarity between the expression profiles of a pair of genes (or groups), and the most commonly used metrics is Euclidean distance (Clatworthy 2005).

Hierarchical clustering provides an easy way of analyzing the similarities in overall gene expression patterns under different experimental treatments, which is based on the pairwise statistical comparison of complete scatterplots rather than individual gene sequences. The data are represented as a matrix of correlation coefficients, which are used to construct a two-dimensional dendrogram for visualization. Despite these advantages, hierarchical clustering has

flaws. Expression patterns of individual gene sequences become less relevant as the clustering process progresses (Sudhher). Besides, an incorrect assignment made early in the process cannot be corrected (Tamayo, Slonim et al. 1999).

2.1.7.2 Principle component analysis (PCA)

Principle component analysis (PCA) is a mathematical algorithm that reduces the dimensionality of the data while retaining most of the variation in the dataset (Jolliffe 2002, Zhao, Gupta et al. 2011). It is commonly used to project high-dimensional data into a low-dimensional space to inspect of the major sources of variation within a dataset. It accomplishes this reduction by identifying directions, called principal components (PCs), along which the variation in the data is maximal. The eigenvalues and eigenvectors are calculated based on the covariance matrix. The eigenvectors are orthogonal and the one with the largest eigenvalue is the best principle component (PC) of the dataset. To reduce the dimension, the components of less significance would be deleted. This is based on the theory that ignoring eigenvectors with small eigenvalues will not lead to much information loss (Smith 2002). The mathematics behind the PCA is as follows. Consider a data set of observations $[x_n]$, where $n = 1, \dots, N$. The sample mean of the data is

$$\bar{X} = \frac{1}{N} \sum_{n=1}^N x_n$$

The variance of the projected data to D-dimensional vector u_1 is

$$\frac{1}{N} \sum_{n=1}^N [u_1^T x_n - u_1^T \bar{x}]^2 = u_1^T S u_1$$

where u_1 is a D-dimensional vector and S is a covariance matrix defined by

$$S = \frac{1}{N} \sum_{n=1}^N (x_n - \bar{x})(x_n - \bar{x})^T$$

To maximize the projected variance $u_1^T S u_1$ with respect to u_1 , we could make an unconstrained maximization of

$$u_1^T S u_1 + \lambda_1 (1 - u_1^T u_1)$$

By setting the derivative with respect to u_1 equal to zero, the quantity above will have a stationary point when

$$S u_1 = \lambda_1 u_1$$

where u_1 and λ_1 are the eigenvector and eigenvalue of S respectively.

$$u_1^T S u_1 = \lambda_1$$

The variance will be a maximum when we set u_1 equal to the eigenvector having the largest eigenvalue λ_1 . Additional principal components are defined by choosing each new direction to be that which maximizes the projected variance among all possible directions orthogonal to those already considered. For example, if the dimension of the original data is m , and then u_1, u_2, \dots, u_m eigenvectors and $\lambda_1, \lambda_2, \dots, \lambda_m$ eigenvalues are calculated. For high-dimensional data with large m , only the top n eigenvectors with large eigenvalues are kept ($n \ll m$). By doing this, each sample is represented by a reduced number of variables (n) compared with the original number of variables (m). The similarities and differences between samples could be visually assessed (Sturn, Quackenbush et al. 2002). Figure 2.3 is an illustration of finding 2 PCs for 2-dimensional data ($m=2$).

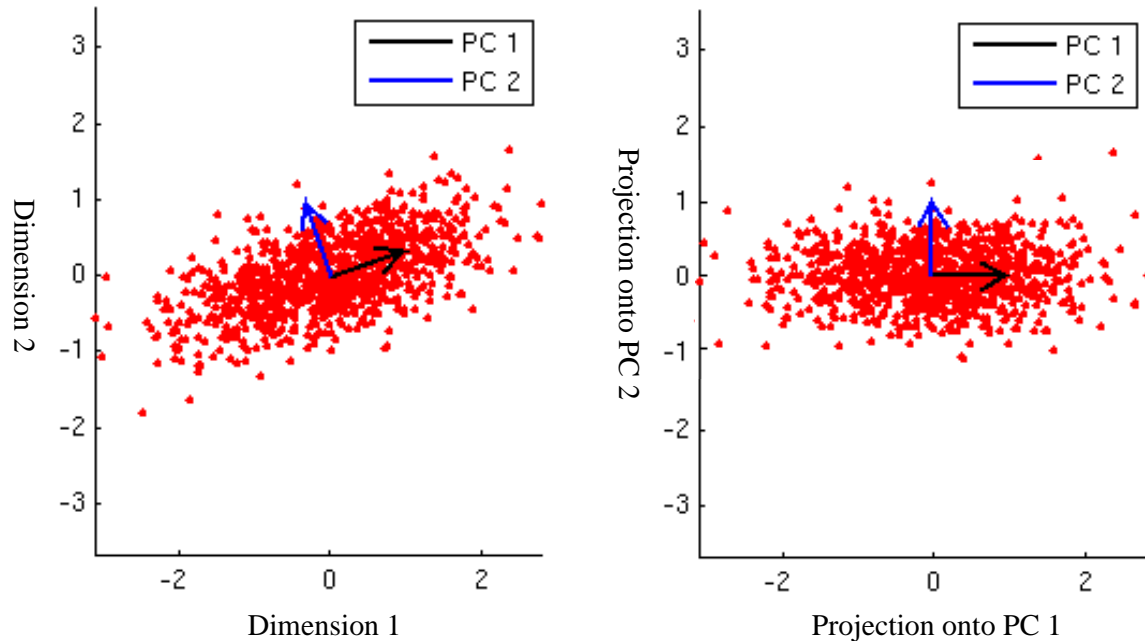


Figure 2.3. Illustration of principle component analysis (PCA).

PCA is often incorporated into genome-wide studies to reduce the dimension of gene expression data and extracts small number of representative features (hidden regulatory signals) that can represent the most information contained in gene expression data (Ringnér 2008). (Ma and Kosorok 2009) used PCA to detect the latent variables that have a strong correlation with a cluster of genes based on their expression patterns under different stimuli.

The shortcoming of the PCA is that the hidden representatives found by it are constrained to be mutually orthogonal and statistically independent (Liao, Boscolo et al. 2003). In an essence, PCA searches for a set of orthogonal eigenvectors that span the space of the data samples, with eigenvectors representing the directions along which data have the biggest variances. However, due to the requirement of orthogonal eigenvectors, the eigenvectors usually cannot capture the biologically meaningful covariance structure of gene expression patterns.

2.1.7.3 Non-negative matrix factorization (NMF)

Non-negative matrix factorization (NMF) is a group of algorithms in multivariate analysis and linear algebra where a matrix V is factorized into two matrices W and H , with the property that all three matrices have no negative elements. NMF is an algorithm based on decomposition by parts that can reduce the dimension of a matrix V (DD Lee , Devarajan 2008).

$$V = W * H$$

Assume that the matrix V is gene expression data. Given this gene expression data, NMF factorizes V into a basis matrix (W) and a coefficient matrix (H). All three matrices should not have any non-negative elements, hence the name non-negative matrix factorization. The number of hidden regulators needs to be pre-defined by a user, and is usually set to a value much smaller than the number of genes. Therefore, NMF helps reduce the dimensionality from thousands of genes to a collection of hidden regulators. Matrix W represents the association between a hidden regulator and a gene. The collection of genes that have strong associations with one hidden regulator could be regarded as a molecular pattern and perform similar functions in the signaling pathway.

NMF was shown to be advantageous over other clustering techniques, such as hierarchical clustering (Carmona-Saez, Pascual-Marqui et al. 2006, Qi, Zhao et al. 2009). Different from PCA whose components needs to be orthogonal and lack intuitive meaning, (Lee 2001) demonstrated that NMF was able to learn localized features with obvious interpretations. Benefiting from the capability of NMF for recognizing the similarity between subsets of the data corresponding to localized features in expression space (Alkim, Benbadis et al. 2013), it has been applied to reduce the dimension of expression data from thousands of genes to a handful of hidden representations (ex. metagenes) (Brunet, Tamayo et al. 2004, Carmona-Saez, Pascual-

Marqui et al. 2006). It is useful for identifying distinct molecular patterns (Brunet, Tamayo et al. 2004). (Kim and Tidor 2003) used it to cluster gene expression profiles and characterize the function of genes. (Gao 2005) applied it to classify different cancer classes/subclasses based on microarray expression data.

However, a cellular signaling system is organized as a hierarchical network such that signaling proteins at different levels compositionally encode signals with different degrees of complexity. For example, activation of the epidermal growth factor receptor (EGFR) leads to a broad change of cellular functions including the activation of multiple signaling molecules such as Ras and MAP kinases (Alberts, Jonson et al. 2008). These signaling molecules activate different transcription factors, eg., Erk-1 and c-Jun/c-Fos complex, with each responsible for the transcription of a subset of genes responding to EGFR treatment. The signals encoded by signaling molecules become increasingly more specific, and they share compositional relationships. Therefore, NMF with only one hidden layer is not able to capture the compositional relationships of cellular signaling transduction system (CSTS). The information represented by the hidden variables of NMF is still complex and hard to interpret.

Besides the methods above, singular value decomposition (Holter, Mitra et al. 2000), independent component analysis (ICA) (Liebermeister 2002, Kong, Vanderburg et al. 2008) and network component analysis (NCA) (Liao, Boscolo et al. 2003) have also been applied to extract a reduced number of features containing biologically significant information from high-throughput datasets.

2.1.8 Discovering signals embedded in transcriptomic data by functional analysis of gene sets

The previous section discusses statistical approaches for identifying statistical structures in gene expression data as a means to discover cellular signals embedded in transcriptomic data *de novo*. In addition to those approaches, researchers also developed different knowledge-based approach to discover the information relevant to cellular signaling from transcriptomic data. This line of research utilizes our knowledge of gene functions to identify sets of differentially expressed genes that perform related functions. The knowledge-based signal discovery approach is based on the assumption that, if a set of functionally related genes is co-differentially expressed under different conditions, these genes are likely regulated by a common signaling pathway in cells. Hence such genes can be treated as the expression signature of pathways (Lu and Lu 2012, Lu, Jin et al. 2013). Functional analysis of transcriptomic data also provides a way for researchers to interpret transcriptomic data. For example, when a set of genes (gene expression signatures) is clustered together by clustering algorithms or highly associated with a particular hidden regulatory unit of NMF, a researcher may want to know the functions of those genes. While functional analysis of transcriptomic data may reveal the signatures of pathways, it does not provide information about which aberrant signaling pathways underlie the changed expression of signatures (Lussier and Chen 2011). Furthermore, these methods depend on existing knowledge of gene (protein) functions, which is incomplete. The following subsections reviews common methods for mapping the gene set to known biological functions and pathways.

2.1.8.1 Gene Ontology (GO) analysis

Gene Ontology (GO) is the framework for the model of biology. It defines both concepts/classes used to describe gene function, and the relationships between these concepts. It classifies functions including molecular function, cellular component and biological process (Ashburner, Ball et al. 2000, Harris, Clark et al. 2004, Khatri and Draghici 2005). GO annotations provides an excellent explanation of the biological relevance of the query genes (Ashburner, Ball et al. 2000). For example, we could use statistical and computational methods to get the differentially expressed genes under specific experimental conditions. When a gene set of interest is available, the GO terms associated with those genes could be obtained by performing GO enrichment analysis. Based on the available GO annotations, we could infer the biological functions/pathways associated with specific stimuli. One of the frequently used GO databases for yeast is the *Saccharomyces* Genome Database (SGD) (Carreira-Perpinan and Hinton 2005).

2.1.8.2 Network-extracted ontology (NeXO)

The Network Extracted Ontology (NeXO) is a gene ontology inferred directly from large-scale molecular networks. It provides structured knowledge about the cellular components, processes, and functions encoded by genes. Like GO database, NeXO uses an ontology of NeXO terms to summarize gene function. Different from GO database that are constructed through manual expert curation, NeXO is a data-driven gene ontology inferred directly from omics data. NeXO is a “shift from using ontologies to evaluate data to using data to construct and evaluate ontologies” (Dutkowski, Kramer et al. 2013). NeXO uses a principled computational approach which integrates evidence from hundreds of thousands of individual gene and protein interactions to construct a complete hierarchy of cellular components and processes (Dutkowski,

Ono et al. 2014). This data-derived ontology aligns with known biological machinery in the GO database and also uncovers many new structures. Therefore, NeXO contains terms and term relations that are not cataloged in GO.

2.1.8.3 Kyoto Encyclopedia of Genes and Genomes (KEGG)

KEGG is a manually curated database resource containing information regarding genes and genome (Kanehisa and Goto 2000). It hosts multiple comprehensive catalogues of genes, proteins, and chemicals. It also contains information of different biochemical and signaling transduction pathways, depicting the relationships between molecules involved in pathways, e.g., enzymes and substrates or signaling molecules in different signaling pathways. KEGG database is commonly used to analyze the transcriptomic data, where a researcher may examine if the members of a pathway are particularly enriched in a set of differentially expressed genes.

2.1.8.4 Gene Set Enrichment Analysis (GSEA)

The GSEA analysis is another popular tool for analyzing if genes involved in different biological function or processes are enriched in a set of differentially expressed genes (Subramanian, Tamayo et al. 2005). It allows a user to choose among different sets of gene signatures, more specifically a list of genes annotated to be involved in a biological process or molecular function. GSEA examines if genes in a given signature is collectively ranked high among differentially expressed genes (according to certain statistical significance metrics) (Subramanian, Kuehn et al. 2007). Another method for gene set enrichment analysis is the hypergeometric testing (Falcon 2008). When a gene set of interest is available, we could use hypergeometric testing to calculate the enrichment p-value between the gene set of interest and the predefined gene set for a specific function. A high enrichment score indicates that gene set of

interest is rich with the predefined gene set, suggesting that the gene set is related to certain biological function. This method works well if the goal is to map the function of a gene set to a pre-defined biological process.

2.2 THE NEEDS FOR NOVEL MODELS FOR STUDYING THE TRANSCRIPTOMIC REGULATION SYSTEM

The previously used latent models, such as PCA and NMF, concentrate on deriving mathematical representations to find low-dimensional hidden units that capture the statistical information (variances and covariances) in high-dimensional data. However, those hidden representations cannot reflect the hierarchical organization of cellular signaling systems because they do not allow hierarchical relationship between latent variables (they are “flat” models). Due to the non-hierarchical structure of PCA and NMF, the information captured by reduced features cannot fully reflect the hierarchical and compositional nature of transcriptomic data, e.g., activation of pheromone receptor activate multiple pathways, each pathways regulated multiple TFs and each TF regulates a set of genes. As such, the information captured by a latent variable in a flat model often retains the convoluted signals, which is hard to interpret.

In order to gain fine-grained information regarding transcriptomic regulation system, it is important to explore models that can capture the hierarchical signaling mechanisms employed by cells in response to different stimuli. We need a model with a hierarchical structure to reflect different degrees of complexity and capture the probabilistic relationship between the biological components involved in the network. This motivated us to investigate whether “deep learning” models can be applied to study cellular signaling systems or not.

2.3 DEEP LEARNING

2.3.1 History of deep learning

Deep learning models (DLMs) are extensions of multiple-layer artificial neural networks (ANNs), which were originally developed in the 1980s (Rumelhart 1988, Hinton 1992). For a couple decades, the ANN models were confronted with overfitting problems that limited its application. They regained popularity due to new inference algorithms developed a decade ago. The availability of large volumes of data also helps to overcome overfitting problems associated with conventional ANNs. Deep learning began to capture researchers' attention after a paper about successfully applying an autoencoder model to classify MNIST digit image was published in *Neural Computation* in 2006 (Hinton, Osindero et al. 2006) and *Science* (Hinton and Salakhutdinov 2006).

DLMs utilize hierarchically organized latent variables to capture the compositional relationship of statistical structures existing in input data (Hinton and Salakhutdinov 2006). This kind of representation is partially inspired by the interpretation of information processing and communication patterns in a nervous system, such as neural coding which attempts to define a relationship between various stimuli and associated neuronal responses in the brain (Le 2013). DLMs have been shown to dramatically improve upon state-of-the-art machine learning methods, such as support vector machine (SVM) (Suykens 1999), regression (Tibshirani 1996) and random forest (Liaw 2002), in image recognition, speech recognition and natural language processing (Collobert 2008). Nowadays, the application of DLMs has broadened into fields such as the biomedical field.

2.3.2 Basic structure of artificial neural networks

The artificial neural network model is inspired by biological neural networks (Hagan 1996, Jain 1996). It is composed of one observed input layer and multiple hidden layers (Figure 2.4). Hidden layers could be regarded as feature detectors of signals embedded in neurons (inputs). To pass the information of neurons from one layer to another layer, the neuron combines the weighted inputs and biases, and then sends it through an activation function (Figure 2.5, section 2.3.2.1) to get the output. The output is then used as input to the next layer. The parameters of the model, including weights and biases, are adjusted through the procedure of backpropagation (section 2.3.2.2).

2.3.2.1 Activation function at each neuron

An artificial neural network uses the activation function to pass the information of neurons from one layer to another layer. Every activation function takes a linear combination value and performs a certain fixed mathematical operation on it (Figure 2.5).

$$f(WX+b)$$

where X is the input matrix whose dimension is the number of samples \times the number of input units. W is the weight matrix whose dimension is the number of input units \times the number of output units. b is the bias vector for output units and f is the activation function.

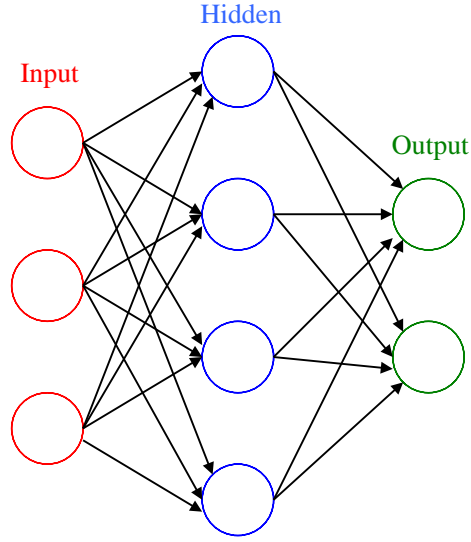


Figure 2.4. Basic structure of an artificial neural network. An artificial neural network is a multi-layer neural network, which contains an interconnected group of nodes. Each circular node represents an artificial neuron and an arrow represents a connection from the output of one neuron to the input of another.

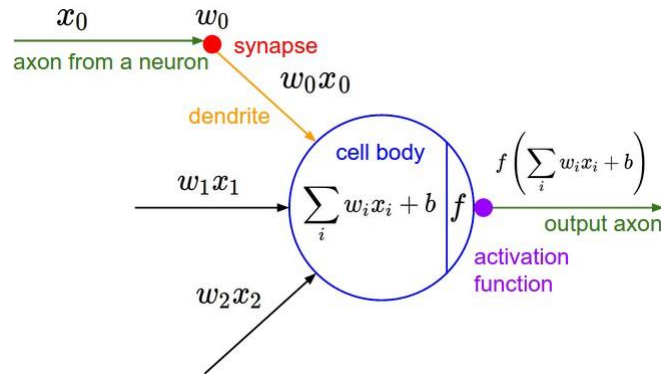


Figure 2.5. Illustration of the activation function of a neural network. Each of the inputs x_i is multiplied by a previously established weight w_i . These are all summed together, resulting in $\sum_i w_i x_i + b$. This value is then biased by a previously established threshold value b , and sent through an activation function f . The resulting output is an input to the next layer.

The following summarizes the three mostly frequently used activation functions for artificial neural networks.

1) Sigmoid function

The sigmoid function is the most frequently used non-linear activation function. The range of the output value is $(0, +1)$ (Figure 2.6). Value 0 means that a neuron is not fired at all and value 1 means that a neuron is completely fired. Firing rate $\sigma(x)$ approaches 0 when x approaches a large negative value and approaches 1 when x approaches a large positive value. The input for the sigmoid function in artificial neural networks is the weighted linear combination of the activation status of neurons in the previous layer as shown in Figure 2.5.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

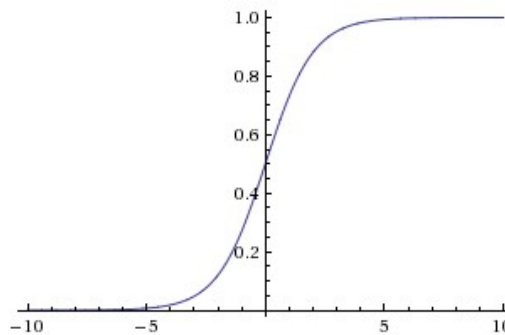


Figure 2.6. Sigmoid activation function.

2) Hyperbolic tangent

The hyperbolic tangent (tanh) activation function is simply a scaled sigmoid activation function (Figure 2.7). The range of the output value is $(-1, +1)$.

$$\tanh x = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

$$\tanh(x) = 2\sigma(2x) - 1$$

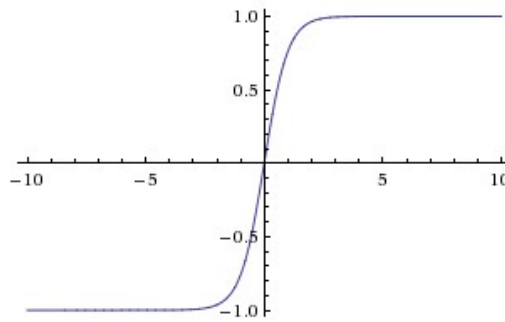


Figure 2.7. Hyperbolic tangent activation function.

3) Rectified linear unit (ReLU)

The rectified linear unit (ReLU) is a linear activation function and has become very popular in the last few years (Nair 2010). It computes the function $f(x) = \max(0, x)$. It was shown to greatly accelerate the convergence of stochastic gradient descent compared to the sigmoid/tanh functions due to its linear non-saturating form (Nair 2010). The range of the output value is $[0, +\infty)$ (Figure 2.8). $f(x)$ is 0 when x is a negative number and is x when x is a positive number.

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

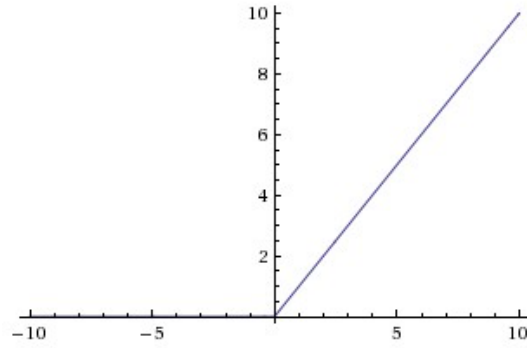


Figure 2.8. Rectified linear unit (ReLU) activation function.

2.3.2.2 Backpropagation to train a multi-layer network

When training a DLM, parameters associated with the activation function of each neuron are randomly initialized which, of course, won't perform well when performing supervised or unsupervised tasks. Backpropagation procedure is a gradient descend algorithm for adjusting model parameters in order to minimize the error between the observed data and model prediction. Training a neural network involves the layer-wise invoking of the activation function from input layer to output layer and then applying a backpropagation gradient descend to adjust model parameters (Figure 2.9). Computing the gradient of an objective function (e.g., squared loss) with respect to the weights of a multilayer stack of modules is achieved through application of the chain rule of derivatives. The key insight is that the derivative (or gradient) of the objective with respect to the input of a module can be computed by working backwards from the gradient with respect to the output of that module (or the input of the subsequent module). The backpropagation equation can be applied repeatedly to propagate gradients through all modules, starting from the output at the top all the way to the bottom (Eq. in A.4.3). Once these gradients

have been computed, parameters can be adjusted following the gradient with a user-defined learning rate.

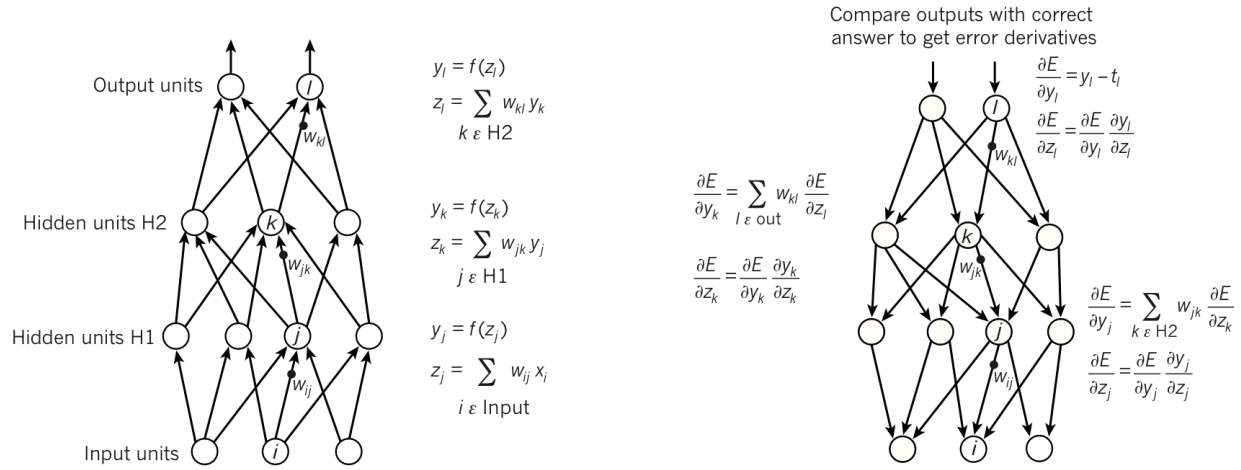


Figure 2.9. Illustration of backpropagation. (LeCun 2015)

2.3.3 Supervised and unsupervised learning

DLMs can be used to perform supervised learning and unsupervised learning. In a supervised learning, a training case usually consists of a set of variables/features (X) and a target variable (Y). The task is to learn a mapping function between X and Y :

$$f(X) \rightarrow Y$$

Given a set of training cases, DLMs formulate a supervised learning task as to use multiple layers of latent variable to extract the statistical structures of the input data and learn a mapping from latent variables to output variables (Figure 2.10A). The mapping is evaluated by the cost function, which is the mismatch between the output $f(X)$ and the target value Y . A commonly

used cost function is the mean squared error, which minimizes the average squared error between $f(X)$ and Y over all the cases.

In an unsupervised setting, a training dataset consists of cases with a set of input variables, and the task is to learn a model that is capable of capturing the statistical characteristic of input cases such that the model can regenerate the training cases. An unsupervised DLM is illustrated in (Figure 2.10B).

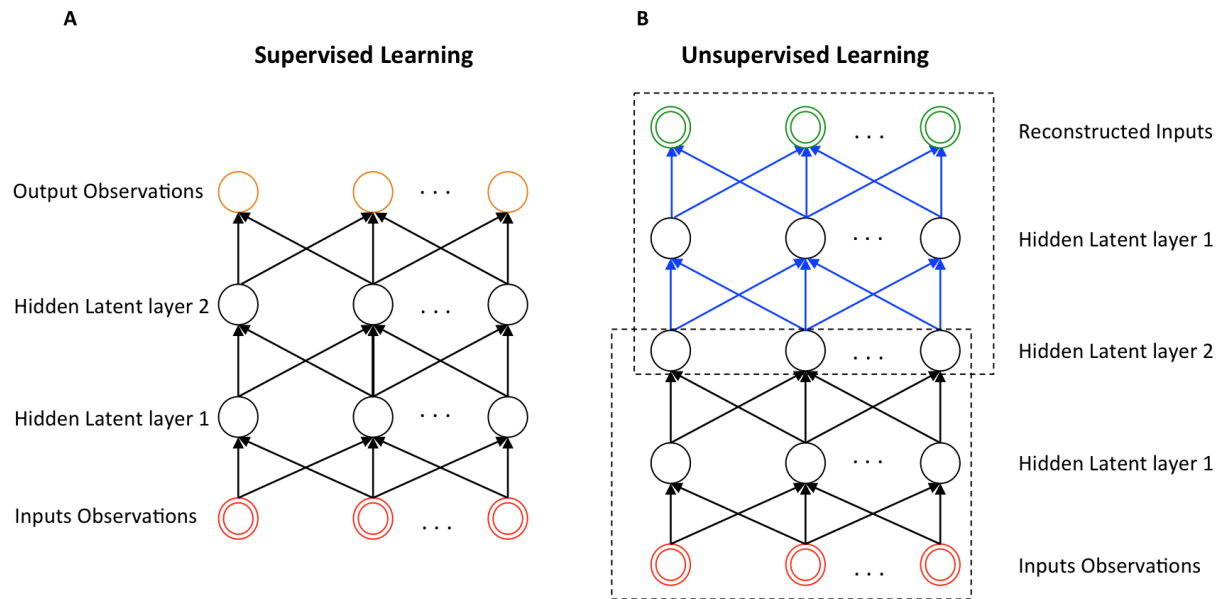


Figure 2.10 Supervised learning and unsupervised learning.

During training and prediction, the information from input variables are progressively propagated through latent variable layers (eventually to the output variable in the case of supervised learning) using an activation function (section 2.3.2.1), and the parameters associated with each latent variable (hereafter referred to as a “neuron”, following the convention of ANN)

is most commonly learned using a backpropagation algorithm (section 2.3.2.2). The details of supervised and unsupervised learning are in Appendix A.1.

2.3.4 Deep learning models

2.3.4.1 Deep belief network (DBN)

The DBN is closely related to an unsupervised neural network model referred to as an autoencoder (Hinton and Salakhutdinov 2006). The autoencoder consists of two parts: the encoder and the decoder. The encoder transforms the high-dimensional data into a low-dimensional code and the decoder recovers the high-dimensional data from the low-dimensional code to reconstruct the input data (Figure 2.11). Therefore, it is suited for solving the unsupervised reconstruction problem (Brown 2005). By adding a classification layer (ex. Softmax layer) above the output layer, the model could perform supervised classification learning (Hinton, Osindero et al. 2006, Krizhevsky 2012) that classifies the input samples into classes. A significant change in the training of the DBN from the conventional training of the deep neural network (DNN) is that the new training algorithm employs a pre-training step to update parameters before backpropagation. During pre-training, a DBN is regarded as a repeated stack of an undirected bipartite probabilistic graphical model called restricted Boltzmann machine (RBM) shown in Figure 2.12 (Appendix A.2). Experimental evidence reported that pre-training could help the subsequent optimization performed by stochastic gradient descent (backpropagation) (Erhan, Bengio et al. 2010, Mohamed, Dahl et al. 2012). The training results of the DBN may be similar to those of DNN when the number of hidden layers is only one or two. However, the advantage of the DBN over DNN becomes apparent when the number of hidden layers increases.

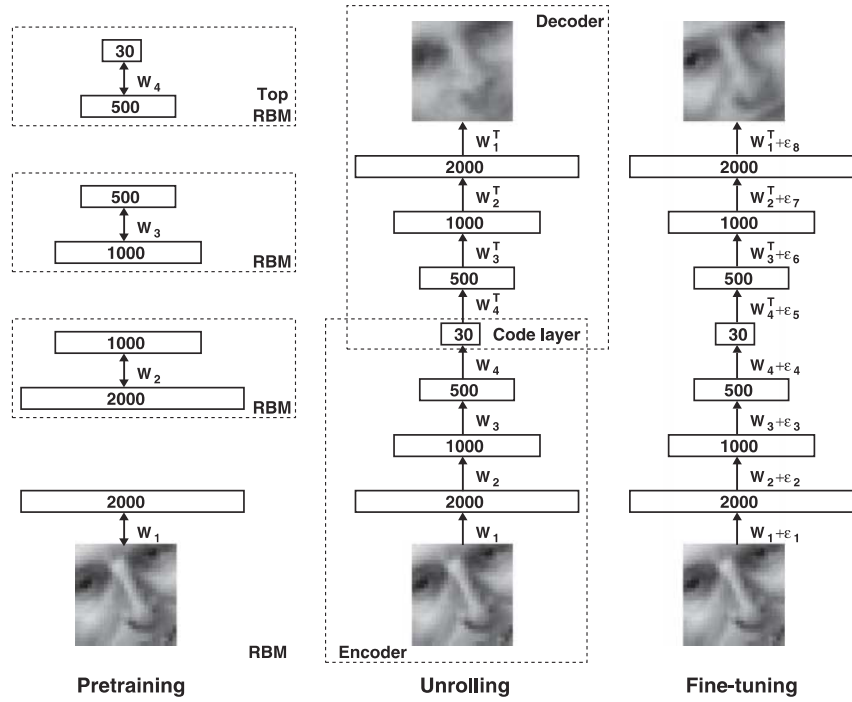


Figure 2.11. Autoencoder. Pretraining consists of learning a stack of restricted Boltzmann machines (RBMs), each having only one layer of feature detectors. The learned feature activations of one RBM are used as the input data for training the next RBM in the stack. After the pretraining, the RBMs are “unrolled” to create a deep autoencoder, which is then fine-tuned using backpropagation of the error derivatives (Hinton, Osindero et al. 2006).

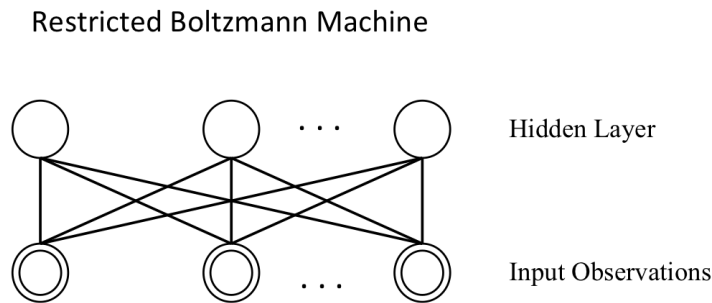


Figure 2.12. Restricted Boltzmann Machine (RBM).

2.3.4.2 Sparse Deep Belief Network

Sparse deep belief network (sparse DBN), similar to DBN, can be regarded as stacked sparse RBMs combined with backpropagation fine-tuning. However, the large number of parameters in DBN could easily lead to the problem of overfitting. Different from a traditional DBN, a sparse DBN uses a regularization approach when performing pre-training. For the sparse RBM, it adds a regularization term into a traditional RBM to restrict the percentage of active hidden units. By adding this regularization, a sparse DBN forces certain hidden units in each layer to be set at the inactive state (0s), thus efficiently reducing the number of non-zero parameters (sparse). It is shown that the sparse autoencoder can partially address the problem of overfitting (H Lee 2008).

2.3.4.3 Dropout Neural Network

Dropout is another approach for mitigating the overfitting problem. A model with dropout randomly drops some hidden units from the neural network during training (Figure 2.13) (Srivastava 2014). By randomly dropping some units, it prevents the units from co-adapting too much. At the test time, it just uses a single no-dropout network with the weights associated with a hidden unit proportional to the dropout rate to approximate the effect of averaging the predictions of networks.

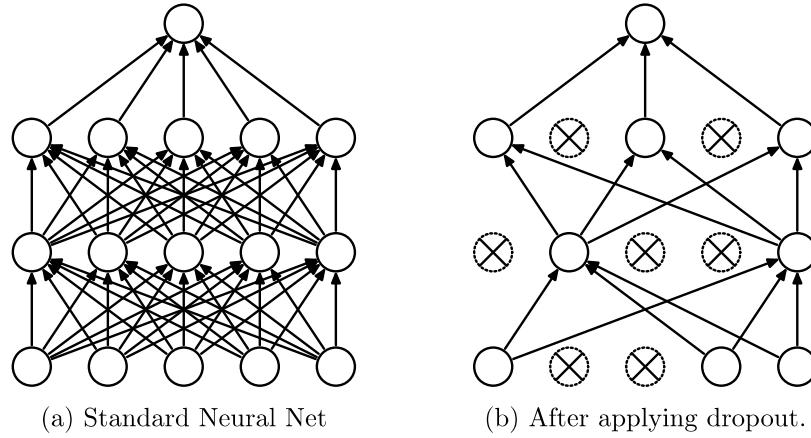


Figure 2.13. Dropout Deep Belief Network. (A) A standard deep neural network (DNN) (B) An example of a thinned DNN by applying dropout to the network on the left (Srivastava 2014).

During training, a unit is present with a constant probability p and is connected to units in the next layer with weights w . For each presentation of each training case, a sampling procedure is involved to determine if a hidden unit should be set to inactive (0), hence the term “drop off”, in a thinned network specific for this case. Forward and backpropagation are performed on the thinned network for each training case. The gradients of parameters are averaged over the training cases in each mini batch. At test time, it’s not feasible to explicitly average the predictions from all thinned models; an approximate averaging method is used instead. The approximate averaging method uses a single neural net during testing without dropout, and the weights between a hidden unit to the next level units are multiplied by p , the probability that a node is being dropped (Srivastava 2014).

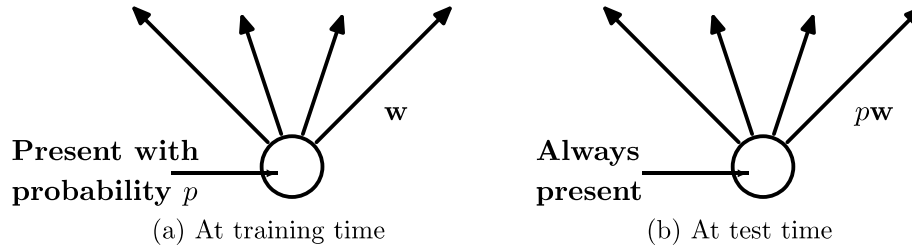


Figure 2.14. Presence of a unit at training time and test time. (A) A unit at training time that is present with probability p and is connected to units in the next layer with weights w . (B) At test time, the unit is always present and the weights are multiplied by p . The output at test time is the same as the expected output at training time (Srivastava 2014).

The DBN with dropout is regarded as an efficient way of averaging models and performing regularization. Dropout was shown to significantly reduce overfitting and had improvements over other regularization methods (Hinton 2012) (Srivastava 2013).

2.3.4.4 Deep Boltzmann Machine (DBM)

The structure of a Deep Boltzmann Machine (DBM) looks pretty similar to a Deep Belief Network (DBN). They all use stacked Restricted Boltzmann Machines (RBMs) for pre-training of parameters. However, they are essentially different because the DBN is a directed model and the DBM is an undirected model. DBNs are sigmoid belief networks with many densely connected layers of latent variables. DBMs are Markov random fields (MRFs), which contain a set of random variables that have a Markov property. DBMs are also composed of many densely connected layers of latent variables (Salakhutdinov 2009).

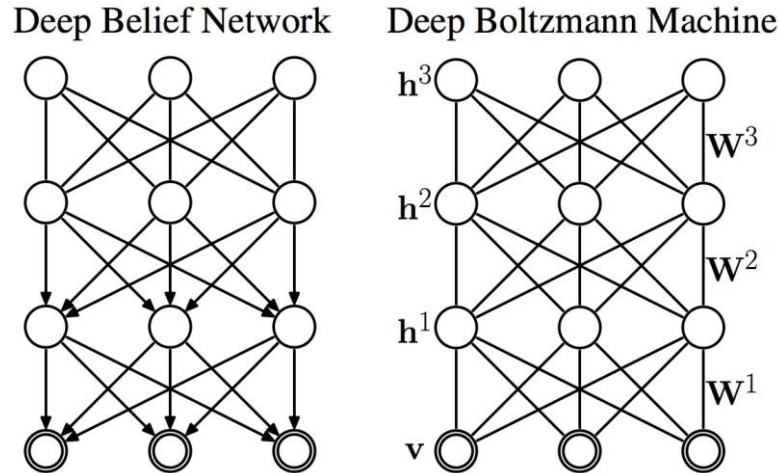


Figure 2.15 A three-layer Deep Boltzmann Machine (DBM) and a three-layer Deep Belief Network (DBN).

2.3.4.5 Multimodal Deep Boltzmann Machine

Some machine learning tasks require learning the joint statistical structure among data of different modalities. For example, one such task is to learn to automatically tag a picture with key words, using a collection of pictures annotated with keywords. The challenge is that words and pictures have distinct modalities (statistical structures) and it is a challenge to use a single deep structure to represent both modalities. Instead of using one input modality, multimodal DBM uses multiple deep networks to capture the statistical structure of input data with distinct modality and learn the unified representation that fuses modalities at more abstract level. Figure 2.16 is an illustration of a multimodal DBM, which learns a joint representation between two input modalities by adding a hidden layer (green part in Figure 2.16) above the top layer of two separate DBMs (blue part and yellow part in Figure 2.16). (Srivastava and Salakhutdinov 2012) successfully used the multimodal DBM to learn a generative model of image and text inputs. The representation learnt by the model captures features that are useful for classification and retrieval. It was shown to outperform SVMs on discriminative tasks.

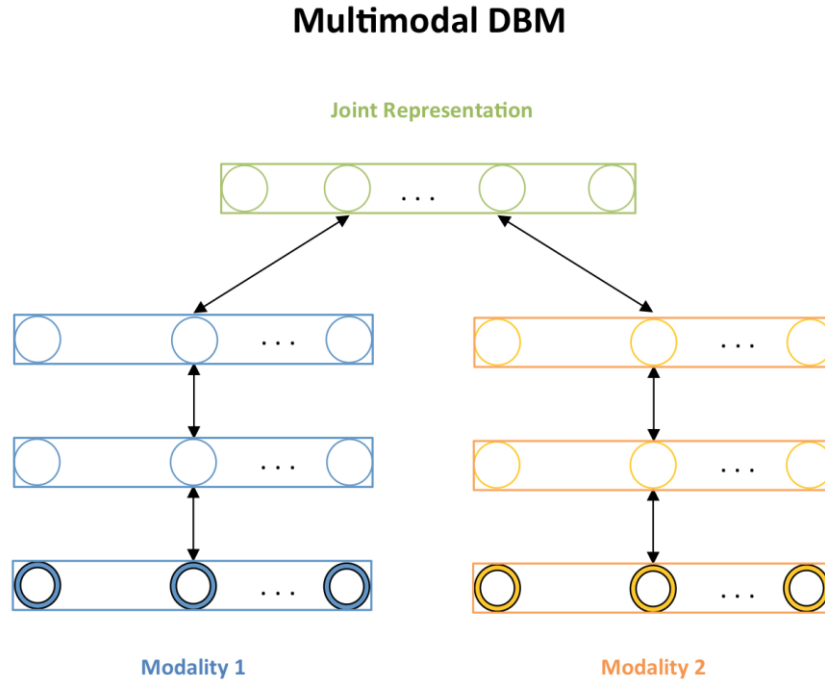


Figure 2.16. Multimodal Deep Boltzmann Machine (DBM).

2.3.4.6 Convolutional Neural Network (CNN)

In image analysis, detecting an object, such as a human face, can be challenging, because such an object can occur in different positions in an image. Besides, different training cases may present the object in different scales and the object can be rotated in different angles. Convolutional neural networks (CNNs) (LeCun 1995, LeCun 1998, Krizhevsky 2012) were developed to address these problems, which aimed to learn a representation of objects insensitive to translation, rotation and scaling of the object. A CNN is comprised of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected pooling layers as in a standard multilayer neural network (Figure 2.17). By setting a different convolution/activation function on different layers of the network, CNN is forced to

learn different features leading to a better result. By applying local connection and tied weights followed by pooling, the translation invariant feature could be successfully found.

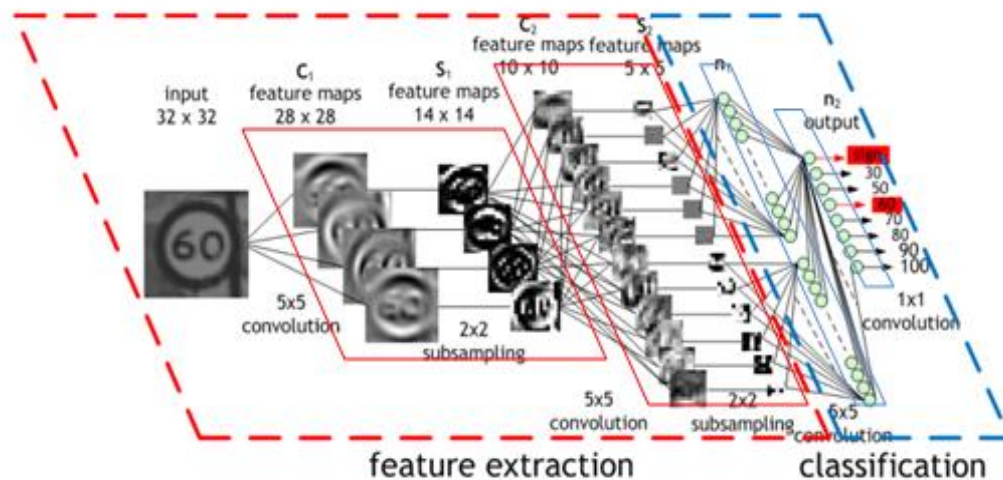


Figure 2.17. Convolutional Neural Network (CNN). (Peemen 2015)

2.3.5 Identification and visualization of information represented by latent variables in hierarchical models

After training a DLM, it is often desirable to learn what statistical structure each hidden node represents. In image analysis, visualization is an important part of evaluating DLMs because it provides a direct view of the statistical structures (parts of images) learned by models training algorithms. The following are several visualization techniques applied in existing deep learning studies. They are 1) weight normalization 2) Weighted linear combination 3) sampling with clamping and 4) activation maximization.

- 1) **Weight Normalization:** The goal is to find an input configuration x that maximally activates a particular hidden unit i in a hidden layer (Ng 2011). For example, (Ng 2011) wanted to visualize how 100 visible units (denoted by x_j) were influenced by the hidden unit i in a hidden layer. They visualized the function of hidden unit i based on the parameter $W_{ij}^{(1)}$ using a 2D image, where $W_{ij}^{(1)}$ is the weight between input visible unit j and hidden unit i . This calculation was performed using norm constraint. For each hidden unit i , each input visible unit was norm-constrained using the equation below:

$$x_j = \frac{W_{ij}^{(1)}}{\sqrt{\sum_{j=1}^{100} (W_{ij}^{(1)})^2}}$$

Figure 2.18 shows the visualization of the function performed by 100 hidden units in the first hidden layer by using the method above.

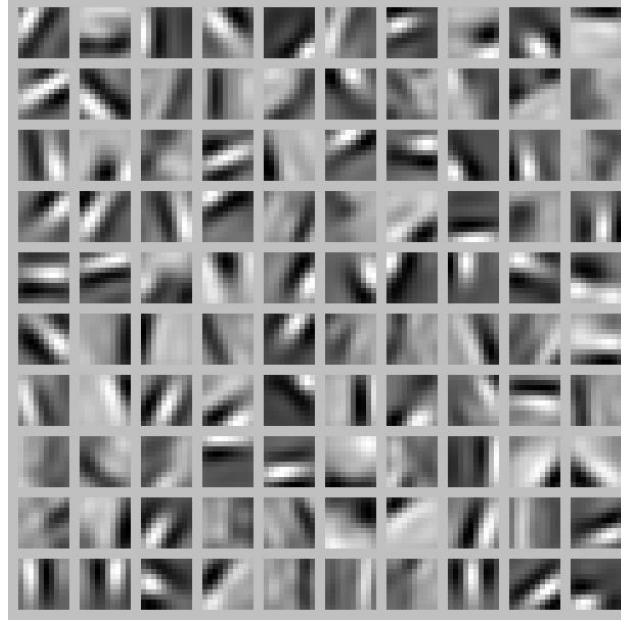


Figure 2.18. Visualization of the first hidden layer by performing weight normalization. Each square in the figure shows the (norm bounded) input image x that maximally activates one of 100 hidden units in the first hidden layer. Different hidden units have learned to detect edges at different positions and orientations in the image (Ng 2011).

- 2) **Weighted Linear Combination:** Weighted linear combination is another method of studying the relationship between the input configuration and a hidden unit in the hidden layer. To get the weight between a particular hidden layer and the visible layer, (H Lee 2008, Le 2013) just linearly combined the weight matrixes between the hidden layer of interest and the visible layer. For example, the weight matrix between the second hidden layer and the visible layer could be calculated with the equation below.

$$w^{h^{1,3}} = w^{h^{1,2}} * w^{h^{2,3}}$$

where $w^{h^{1,2}}$ is the weight matrix between the visible layer and the first hidden layer;
 $w^{h^{2,3}}$ is the weight matrix between the first hidden layer and the second hidden layer;

$w^{h^{1,3}}$ is the weight matrix between the second hidden layer and the third hidden layer. Then they use the weight normalization method discussed above to calculate the contribution of each input unit to the activation of a hidden unit.

The advantages of this method are that it's simple and efficient. The disadvantages of this method are: 1) Instead of a small group of large weights, there may be many smaller and similar-magnitude weights contributing to the activation of a unit. 2) It ignores the nonlinearity between layers.

- 3) **Sampling with Clamping:** This idea was first described by (Hinton, Osindero et al. 2006). Hinton et al. used this method to visualize output class-label units as distributions in the input space. They clamped the label vector (label k from 0 to 9) to a particular configuration and sampled from a particular class distribution $p(x|\text{class} = k)$. A distribution denoted by $p(x|h_{ij} = 1)$ was used to reveal the relationship between h_{ij} and input units x , where h_{ij} is a hidden node i in the hidden layer j . To calculate this distribution, they needed to successively sample between two neighboring layers h_{j-1} and h_j . During this process, they clamped h_{ij} and kept setting this unit to 1. Then they sampled inputs x by performing top-down sampling in the DBN going from layer $j - 1$ to the first hidden layer. Finally, the input units could be characterized by samples from the $p(x|h_{ij} = 1)$ or by $E[x|h_{ij} = 1]$.

The advantage of this method is that the only hyper-parameter is the number of samples used to calculate $E[x|h_{ij} = 1]$. The disadvantage of this method is that sometimes it is really difficult to obtain samples that cover the RBM distribution well.

- 4) **Activation Maximization:** This method looks for input patterns that maximize the activation of a particular hidden unit (Erhan 2009) and helps to understand

representations learnt in deep architectures. The assumption of this method is that “a pattern to which the unit is responding maximally could be a good initial representation of what a unit is doing” (Erhan 2009). Therefore, the task becomes to find the input samples that give rise to the highest activation of a given hidden unit. (Erhan 2009) regarded the activation maximization of a unit as an optimization problem. The parameters θ (weights and biases) are fixed and the activation of the units changes with different input x . The task here is to find the configuration of x that maximizes the activation of h_{ij} .

$$x^* = \underset{x \text{ s.t. } ||x|| = \rho}{\operatorname{argmax}} h_{ij}(\theta, x)$$

where θ represents parameters including weights and biases, and $h_{ij}(\theta, x)$ is the activation of a given unit i from a given hidden layer j in the network corresponding to a particular configuration of input x . Gradient descent is one of the easiest ways of finding the optimization point. The author computed the gradient of $h_{ij}(\theta, x)$ and moved x in the direction of the gradient until it converged. The shortcoming of activation maximization is that it is not clear how to choose the samples kept for the given hidden unit and how to “combine” the information in samples. Therefore, the commonality among samples needs to be explored.

Figure 2.19 is an example of visualizing the function of hidden units in a particular layer by using linear combination, sampling and activation maximization separately with MNIST (top row) and Natural Image Patches (bottom row) as input (Erhan 2009).

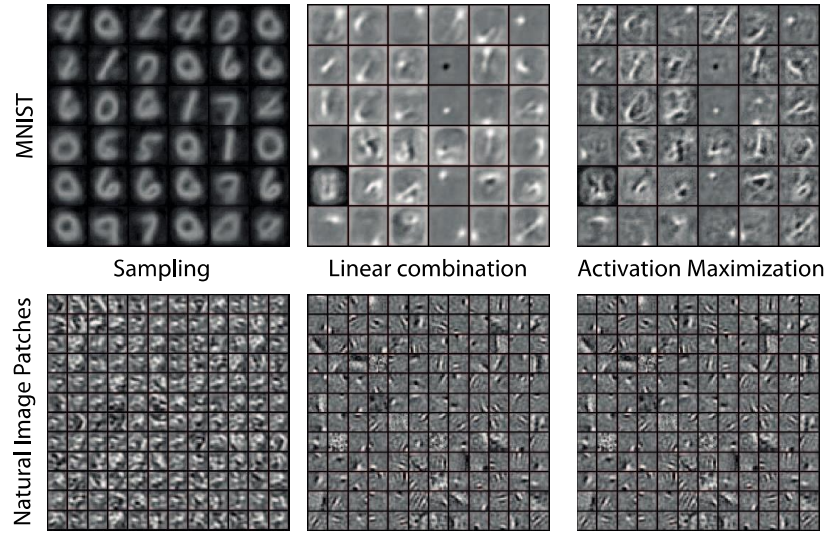


Figure 2.19. Comparison of three visualization methods. The figure from the left to right is 1) sampling with clamping 2) weighted linear combination 3) activation maximization (Erhan 2009)

2.3.6 Application of DLMs in different machine learning domains

DLMs have been intensively studied and proved to improve the state-of-the-art in the following machine learning domain of 1) computer vision or image analysis, 2) speech recognition, and 3) nature language processing.

Various DLMs, such as convolutional deep neural networks and deep belief networks, have been applied to the field of computer vision to perform the task of image processing (LeCun 1995, Hinton, Osindero et al. 2006, Lee 2009, Y. 2009). An image can be represented in many ways such as a vector of intensity values per pixel, or in a more abstract way as a set of edges, regions of a particular shape, object parts etc. DLMs use multiple processing layers to capture information at different abstractions. Figure 2.20 is an example of how DLM models high-level abstractions in image data with multiple processing layers. The first processing layer

could represent a set of edges. The second and third processing layer could represent regions of a particular shape and object parts separately. The higher the layer, the more abstract the information represented by hidden nodes is. By training DLMs with a large amount of image data of various object categories, a new image could be classified into a particular category, such as cat/human face (Le 2013).

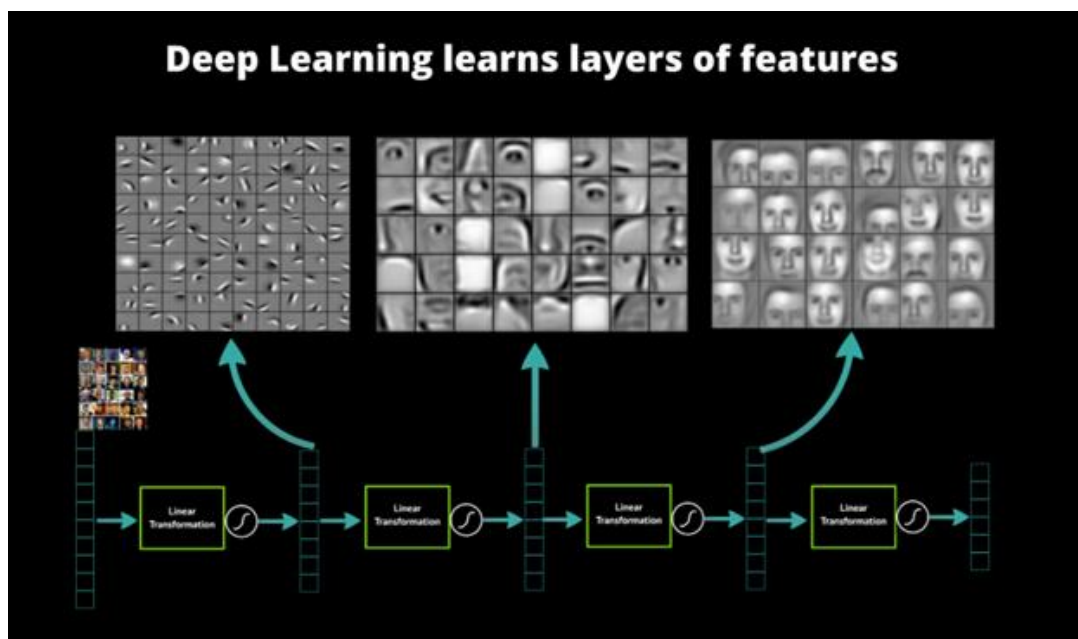


Figure 2.20. Application of DLMs in computer vision. DLMs model high-level abstractions in image data by using multiple processing layers (non-linear transformations)

(<https://s3.amazonaws.com/datarobotblog/images/deepLearningIntro/013.png>).

DLMs were also successfully applied to perform the task of natural language processing and speech recognition (Ronan C. 2008, Hinton G. 2012). For natural language processing, DLMs also use multiple processing layers to model high-level abstractions. The lowest

processing layer could represent the most specific information (word syntax), such as nouns and adjectives. The second layer could represent complex information, such as a noun phrase (NP), which is a combination of an adjective and a noun. Then the higher layer could represent more complex information such as a combination of a verb phrase (VP) and NP. A VP at higher layers could be a combination of smaller VPs and NPs (ex. quietly enters (VP) the historic church (NP)). Various DLMs, such as DBN with multitask learning, have been successfully applied to perform the task of natural language processing (Collobert 2008). For the speech recognition, a convolutional deep belief network that could capture translation-invariance was often used for audio recognition and classification (H Lee 2009).

Even though vision, audio and language have distinct properties, they could represent similar information or concepts (Figure 2.21). After studying them separately, researchers became curious about whether data of different modalities could be combined as input to discover new features. Then, a DLM called multimodal deep Boltzmann machine (multimodal DBM), was developed to learn the joint statistical structure among data of different modalities. (Srivastava and Salakhutdinov 2012) successfully used the multimodal DBM to learn a generative model of image and text inputs. The representation learnt by the model captures features that are useful for classification and retrieval.













Image	Given Tags	Generated Tags	Input Text	2 nearest neighbours to generated image features	
	pentax, k10d, kangarooisland, southaustralia, sa, australia, australiansealion, 300mm	beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves	nature, hill scenery, green clouds		
	<no text>	night, lights, christmas, nightshot, nacht, nuit, notte, longexposure, noche, nocturna	flower, nature, green, flowers, petal, petals, bud		
	aheram, 0505 sarahc, moo	portrait, bw, blackandwhite, woman, people, faces, girl, blackwhite, person, man	blue, red, art, artwork, painted, paint, artistic surreal, gallery bleu		
	unseulpixel, naturey crap	fall, autumn, trees, leaves, foliage, forest, woods, branches, path	bw, blackandwhite, noiret blanc, biancoenero blancoynegro		

Figure 2.21. Similar information embedded in image and text.(Srivastava and Salakhutdinov 2012)

2.3.7 Application of DLMs in biomedical fields

In the era of big data, extracting knowledge from a large quantity of “big data” becomes important in various domains and the biomedical field is no exception. Significant amounts of biomedical data, such as omics data, are collected. Machine learning methods, such as regression, support vector machine, and principle component analysis, have been used to dig out information embedded in biomedical data for decades (Brown, Grundy et al. 2000). Nowadays, deep learning gains attention rapidly due to the power of parallel and distributed computing. The “big” biomedical data leads to the potential for applying DLMs in the biomedical field. Figure 2.22 shows that published papers using deep learning in the biomedical field has grown rapidly since the early 2000s (Min 2016). The following summarizes some main applications in the biomedical field.

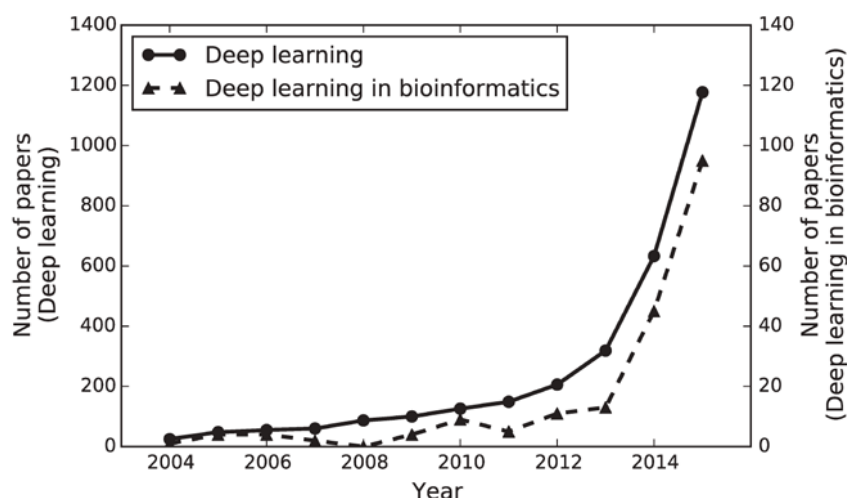


Figure 2.22. Number of deep learning papers in all fields vs. bioinformatics field in the past decade.

(Min 2016)

- **The application of DLMs in diagnosing biomedical-image**

It is well known that deep learning methods were successfully applied in the area of image classification (Hinton, Osindero et al. 2006, Y. 2009, Krizhevsky 2012, LeCun 2015). Recent studies in biomedical image analysis (Suk and Shen 2013, Plis, Hjelm et al. 2014, Esteva 2015) used DLMs to tackle biomedical problems including disease and cancer classification based on images (ex. MRI). One of the studies (Payan 2015) used the combination of a sparse Autoencoder and 3D convolutional neural network to predict the Alzheimer' disease status of a patient based on MRI images.

- **The application of DLMs in predicting DNA- and RNA-binding proteins**

Knowing the sequence specificities of DNA- and RNA- binding proteins is essential for developing models of regulatory processes in biological systems and for identifying causal disease variants. The study (Alipanahi, DeLong et al. 2015) used a convolutional neural network

to model sequence specificities from experimental data for pattern discovery. The advantage of the convolutional neural network over the traditional neural network is that the convolutional neural network could discover new patterns even when the locations of patterns within sequences are unknown. For the traditional neural network, it needs a large amount of training data to do this.

- **The application of DLMs in predicting tissue-regulated PSI (percentage spliced in)**

A deep belief network (DBN) was used to predict splicing patterns based on both RNA-Seq data and tissue type data (Leung, Xiong et al. 2014). The RNA-Seq data can predict splicing patterns in individual tissues and differences in splicing patterns across tissues. The DBN applied by this study used the two data types mentioned above as input and had a hidden layer capturing the joint representation in genomic sequences and tissue types. The prediction of PSI was made based on this joint representation.

- **The application of multimodal DLMs in analyzing ovarian and breast cancer data**

Molecular phenotype readout of signaling transductions may not be limited to gene expression. Other data types, such as mutation data and DNA methylation data, could also be used as phenotype readout. Studies showed that more information about biological components could be captured when a combination of data types instead of a single data type is used (Zhang, Zhang et al. 2013). A multimodal DLM, which consisted of three layers, was used to classify cancer subtypes with the combination of gene expression, DNA methylation and drug response data as input (Liang, Li et al. 2015). The bottom two layers constituted three separate RBMs, which took gene expression, DNA methylation and drug response respectively as input data. The joint hidden layer on top of the 3 separate RBMs captured the joint representation of three data

types. Survival time was used to investigate the discrepancy among the subtypes identified by the multimodal DBN.

- **The application of DLMs in inferring gene expression**

A deep learning model called D-GEX, which is a multi-task multilayer feedforward neural network, was applied for inferring the expression of target genes from the expression of around 1000 “landmark” genes (Chen, Li et al. 2016). The cost of getting 1000 gene expression profiles is much lower than the cost of getting whole-genome expression profiles. The results showed that D-GEX, combined with the dropout regularization technique, performed the best (compared with linear regression (LR) and k-nearest neighbor (KNN)) even when training and prediction were performed on datasets obtained from different platforms (microarray vs. RNA-Seq).

- **The application of DLMs in discovering the cellular signaling transduction system**

The cellular signal transduction system (CSTS) plays a fundamental role in maintaining homeostasis of a cell, and perturbations of the CSTS lead to diseases such as cancers and diabetes. Most cellular signaling pathways eventually regulate gene expression, thus the gene expression data can be used as a readout of the state of the CSTS. However, the information embedded in the gene expression profile is a mixture of responses to different signaling pathways, which is complex. (Chen, Cai et al. 2016) applied the DBN and sparse DBN to deconvolute the signals embedded in the gene expression data. Different layers of the models represent information of different abstraction. For example, the lower hidden layer could represent transcription factors and the higher hidden layer could represent signaling pathways.

2.3.8 The potential of DLMs for drug sensitivity prediction, cancer subtype classification and personalized medicine

- Drug sensitivity prediction

DLMs could be applied to predict drug sensitivity using gene expression as a feature. Gene expression data could be used as input to train the deep learning model with the highest layer to be the classification layer with softmax regression as a supervised learning algorithm. Each node in the softmax layer would represent a particular drug. If the gene expression of a patient is sensitive to a drug, the status of the drug node corresponding to that patient would be 1. Otherwise, it would be 0. This problem could be regarded as a supervised classification problem.

- Cancer subtype classification

By learning hierarchical representations, DLMs could represent the original high-dimensional input data using hierarchical low-dimensional hidden variable data. Therefore, DLMs are regarded as one of the efficient ways of performing dimension reduction. For biomedical data such as gene expression data, there are tens of thousands of gene features. If we cluster the tumor patients based on the original high-dimensional data, the clustering pattern is noisy and not clear. The cancer subtypes could not be well defined. However, if we used the dimension-reduced hidden variable data to cluster cancer patients, the pattern would be clearer (Antoniadis, Lambert-Lacroix et al. 2003). The patients clustered together may not have the same site of origin, which may be due to the fact that similar signaling pathways are shared. If a new pattern is found, it will lead to a new classification of cancer subtypes.

- Personalized medicine

Another promising application of DLMs in the biomedical field is the personalized medicine. Studies (Nguyen 2002, Dettling and Buhlmann 2003, Bloom, Yang et al. 2004) show that even

though patients may share same site of tumor origin such as lung cancer and breast cancer, the signaling pathways perturbed may be different. What's more, cancer patients with different sites of origin may share similar perturbed signaling pathways. Therefore, it now has been controversy of treating cancer patients just based on the tumor type. Deep learning is one of the methods of identifying the specific signaling pathways disturbed in a patient. This knowledge could be applied to understand disease mechanism of cancers. Besides, as we talked above, DLMS has the potential to be applied to predict the drug sensitivity using patient's personalized genome-scale data. With the availability of those personalized information, specific treatment strategy could be designed to cure cancer patient not only based on the tumor types (site of origin), such as lung cancer and breast cancer, but also based on the specific signaling pathways perturbed.

3.0 INTRODUCTION OF STUDIES

Chapter 1 studied the relationship between lipidomic and transcriptomic data. We used Pearson correlation analysis and a regularized regression model to measure the linear correlation between ceramide species and genes. Based on the transcription factor analysis and gene ontology analysis, we found the different set of functionally related genes associated with distinct ceramide species. However, the relationship between the lipids and genes in living organisms is not linear. There are many biological components, such as TFs, involved in signaling transduction from lipids to genes. Useful information may not be fully detected if we only use linear models. More complex non-linear models are needed to represent the hierarchical signaling pathway.

Chapter 2 studies trans-species learning of cellular signaling transduction. Given the rat and human protein phosphorylation training data, we predicted human protein phosphorylation data from the test rat protein phosphorylation data. Instead of using a linear model to study the relationship between proteomic data and transcriptomic data, we developed DLMs including a bimodal deep belief network and a semi-restricted bimodal deep belief network to represent the common hierarchical signal-encoding mechanism between rat and human. The advantage of this hierarchical model over a non-hierarchical one is that “deep learning” models include hierarchically organized latent variables capable of capturing the statistical structures in the observed proteomic data in a distributed fashion. The results show that the models significantly outperform two current state-of-the-art classification algorithms (SVM and Elastic Net). Our

study demonstrated great potential in using deep hierarchical models to simulate cellular signaling systems.

Chapter 3 investigated the utility of contemporary deep hierarchical models to learn a distributed representation of statistical structures embedded in yeast transcriptomic data. Different from the study in chapter 2, which was to use deep learning models for prediction, chapter 3 used the deep hierarchical models to reconstruct the input data and studied what kind of biological components or signals each hidden unit could represent from the aspect of representation learning. We showed that such a model was capable of learning biologically sensible representations of the data and revealing novel insights regarding the machinery regulating gene expression. We anticipate that such a model can be used to model more complex systems, such as perturbed signaling systems in cancer cells, thus directly contributing to the understanding of disease mechanisms in translational medicine.

4.0 CHAPTER 1: DISTINCT SIGNALING OF CERAMIDE SPECIES IN YEAST REVEALED THROUGH SYSTEMATIC PERTURBATION AND SYSTEMS BIOLOGY ANALYSES

4.1 INTRODUCTION

Ceramides constitute a family of structurally related molecules that form the core structure of the broader family of bioactive lipids found in all eukaryotes, the sphingolipids (Hannun and Obeid 2008). These structural variants of ceramide arise from the condensation of one or more sphingoid bases and several fatty acids. These, in turn, can be modified by the addition of distinct hydroxyl groups on either the sphingoid backbone or the fatty acid. Thus, the biosynthesis of ceramides is the product of the combinatorial action of multiple enzymes that control the structural variations of the ceramide products. In yeast (*Saccharomyces cerevisiae*), ceramide biosynthesis (Figure 9.4) generates more than 30 distinct species that can be identified by contemporary mass spectroscopy-based lipidomic approaches (Kolter 2011); in mammals, the total number of ceramide species may exceed 200 (Y. A. Hannun 2011). In humans, ceramides are collectively involved in physiological processes, such as growth regulation and apoptosis, and in pathological conditions, such as diabetes and cancer (Kolter 2011). However, a fundamental question of ceramide-mediated signaling is whether the structural diversity of ceramides underlies functional diversity. That is, do the distinct ceramides encode specific

signals? Although manipulation of individual enzymes of ceramide metabolism has enabled assignment of specific functions to these enzymes (Hannun and Obeid 2008, Mullen, Spassieva et al. 2011, Spassieva, Rahmaniyan et al. 2012), these approaches do not clearly delineate the specific lipid species involved in the process, because sphingolipid metabolism constitutes a highly connected network such that perturbing the function of an enzyme can lead to broad changes in sphingolipid species beyond the substrates and products of the enzyme (metabolic ripple effects) (Cowart, Shotwell et al. 2010, Y. A. Hannun 2011). Pinpointing the functions of the lipid or lipids implicated by manipulating a sphingolipid metabolic enzyme is critical in deciphering the specific downstream pathways and the mechanisms that mediate the changes in cellular behavior, because it is the lipid product and not the enzyme per se that propagates the downstream signal. Therefore, new tools and approaches capable of delineating connections between specific ceramide structures and diverse downstream signaling pathways are needed. *S. cerevisiae* has emerged as a powerful model to dissect metabolic and functional pathways of sphingolipids. Activation of de novo sphingolipid synthesis is essential for yeast to survive heat stress (Wells, Dickson et al. 1998, Cowart, Okamoto et al. 2003, Cowart and Hannun 2007), and sphingolipids mediate specific downstream processes in response to heat stress, such as cell cycle arrest (Cowart, Okamoto et al. 2003, Matmati, Kitagaki et al. 2009), mRNA sequestration (Cowart, Gandy et al. 2010), and inhibition of nutrient uptake (Guenther, Peralta et al. 2008). Microarray analysis revealed that de novo synthesis of sphingolipids mediates the regulation of several hundred genes in response to heat stress (Cowart, Okamoto et al. 2006). This simultaneous sphingolipid-dependent regulation of diverse processes provides an opportunity to identify functions of diverse ceramide species, but also requires the development and application of novel methodology.

4.2 MATERIAL AND METHODS

Yeast strains and culture conditions (Done by wet lab collaborators)

Yeast strains used in this study including genotypes are listed in Supplemental files. YPD (yeast extract, peptone, and dextrose) medium was used for the heat stress experiment, and for fatty acid treatment, synthetic complete (SC) medium containing 0.17% yeast nitrogen base (US Biological), 0.5% ammonium sulfate, 2 mM sodium hydroxide, and 0.07% SC supplement was used; SCD is SC containing 2% dextrose. SCD dropout medium lacking uracil was used in cells transformed with pYES2 plasmid, and SC with galactose lacking uracil was used to induce YDC1 open reading frame in pYES2 plasmid for the overexpression studies. For all experiments, cells were treated during mid-log growth at 30°C. Heat stress was performed by shifting cells to a 39°C water bath after 45 min of pretreatment with myriocin (Sigma) or vehicle. Cultures were harvested by centrifugation at 3000g for 3 min and stored at –80°C. For spot tests, compounds required for specified treatments including glycerol, sodium chloride, acetic acid, caffeine, Congo red, hygromycin B, and Rose Bengal were purchased from Sigma. All compounds, including fatty acid (myristate or oleate) or vehicle (0.1% ethanol), were dissolved in medium by warming to 50°C for 10 min. After medium was re-equilibrated to room temperature, it was mixed with 2× agar at 50°C to make SC with 2% agar; 25 ml of medium was poured into 100-mm petri dishes. Solidified plates were dried for 20 min at 37°C before use. Mid-log cultures in SCD were diluted [OD₆₀₀ (optical density at 600 nm), 0.3], and then 5 µl of four serial 1:10 dilutions was spotted and incubated at 30°C for 3 to 5 days.

Heat stress and ISP1 treatment (Done by wet lab collaborators)

Cells grown to mid-log (OD₆₀₀, 0.6) from overnight cultures were pretreated with 5 µM ISP1 or vehicle (0.1% methanol) for 45 min, and then heat stress samples were shifted from 30°

to 39°C for 15 min. Samples (100 ml) were divided into 10- and 90-ml aliquots for microarray and lipidomic analysis, respectively, and then harvested at room temperature by centrifugation at 3000g for 3 min and flash-frozen in a dry ice methanol bath.

Myristate treatment (Done by wet lab collaborators)

Cells grown to mid-log (OD600, 0.6) from overnight cultures were pretreated with 5 μ M myriocin or vehicle (0.1% methanol) for 45 min and then treated with 1 mM myristate (Sigma) or fatty acid vehicle (0.05% ethanol) for 15 min. Samples (100 ml) were divided into 10- and 90-ml aliquots for microarray and lipidomic analysis, respectively, and then harvested at room temperature for 3 min and flash-frozen in a dry ice methanol bath.

Systematic perturbations and collection of lipidomic and microarray data (Done by wet lab collaborators)

Yeast cells (JK9-3da) were subjected to the following combinations of perturbations: (i) control condition; (ii) ISP1 treatment at 30°C; (iii) heat stress; (iv) heat stress plus ISP1 treatment; (v) control condition for fatty acid supplement experiment, 30°C in SC medium; (vi) myristate treatment at 30°C; and (vii) myristate plus ISP1 treatment at 30°C. Experiments were repeated three times under each of the above condition. RNA was extracted from 108 cells with the hot acid phenol method (M. A. Collart 1993). Synthesis of complementary DNA (cDNA), in vitro transcription labeling, and hybridization onto the Yeast2.0 chip were conducted with the Affymetrix GeneChip Kit. Cells were grown, treated, and extracted, and total protein was measured, all according to (Montefusco, Newcomb et al. 2012), and relative lipid concentrations were quantified according to the method of (Gruhler, Olsen et al. 2005) and normalized to total protein.

YDC1 experiments (Done by wet lab collaborators)

Wild type (BY4741) or *ycd1D* was used to perform experiments. Growth conditions and heat stress were done as described above. To achieve YDC1 overexpression, a BY4741 strain was transformed with pYES2 plasmid containing an open reading frame of YDC1 under galactose promoter. For galactose induction, cells were harvested from a dextrose-containing medium by centrifugation at 3000g for 3 min; pellets were washed with sterile water, then inoculated into a galactose-containing medium, and grown for 6 hours before treatment with heat stress. After heat stress, cells were harvested by centrifugation, washed with sterile water, and centrifuged again. The pellets were snap-frozen in liquid nitrogen until ready for RNA extraction.

Quantitative real-time reverse transcription polymerase chain reaction (Done by wet lab collaborators)

Total RNA was harvested with hot acid phenol method, described in Short Protocols in Molecular Biology, unit 13.10 (Ausubel. 2002). First-strand cDNA was produced as described previously (Kitagaki, Cowart et al. 2009). Real-time analysis was done with 7500 Real-Time PCR System (Life Technologies), and SYBR Green Supermix protocol (Bio-Rad) was used to perform the analysis. Primers used in the reverse transcription polymerase chain reaction are AFT1 forward primer (TCAAAAGCACACATTCCCTCA) and AFT1 reverse primer (AACTTTAAATGCGTCCGACC). The expression of target genes was normalized to the expression of three reference genes: RDN18, ALG9, and TAF10. The primers are as follows: RDN18, CCATGGTTTCAACGGGTAACG (forward) and GCCTTCCTTGGATGTGGTAGCC (reverse); ALG9, CACGGATAGTGGCTTTGGTGAACAATTAC (forward) and TATGATTATTCTGGCAGCAGGAAAGAACTTGGG (reverse); TAF10, ATATTCCAGGATCAGGTCTTCCGTAGC (forward) and GTAGTCTTCTCATTCTGTTGATGTTGTTGTTG (reverse).

Ontology-based gene function analysis

Given a set of genes that are significantly correlated to a specific lipid species and their annotations in the form of the GO (Ashburner, Ball et al. 2000) (<http://www.geneontology.org>) terms, we aimed to group genes into nondisjoint subsets, such that each module contained genes with closely related GO annotations, and the overall function of the module was represented by a GO term that captured most of the semantic information of the original GO annotations of the genes. We represented genes and their annotations with a data structure referred to as GOGene graph (Muller, Richards et al. 2009, Chen and Lu 2013). In such a graph, a node represents a GO term, and a directed edge between a pair of nodes reflects an “is a” (ISA) relationship between the GO terms, that is, parent term subsumes that of the child term. In addition, each node kept track of the genes it annotated; therefore, the graph contained information of both GO terms and genes. We constructed a canonical graph with all GO terms in the Biological Process namespace, according to the ontology definition from the GO consortium (<http://www.geneontology.org>). When given a set of genes and their annotations, we associated the genes to GO terms on the basis of their annotations, and then we trimmed leaf nodes that had no genes associated. This produced a subgraph in which leaf nodes were a subset of the original GO annotations associated with the genes of interest. Under such a setting, the task of finding functionally coherent gene modules can be achieved by grouping genes according to their annotations through collapsing GOGene graph in a manner that leads to minimal information loss, and we stopped merging when the P value of assessing the functional coherence of a gene module was equal or greater than 0.05 (Chen and Lu 2013).

Microarray and data analysis

Affymetrix CEL files of the microarray experiments were processed with the “affy” package (v 1.24.2), and differential expression was assessed with the “limma” package (v 3.2.3) of the Bioconductor Suite (<http://www.bioconductor.org/>). The threshold for detecting differential expression was set at $P < 0.01$ and $q < 0.05$.

Consensus clustering of lipidomic data

The R implementation of the clusterCons (20) was downloaded from the CRAN (Comprehensive R Archive Network) (<http://cran.r-project.org/web/packages/clusterCons/>). Lipidomic data (32 species in 21 experimental conditions) were used as input for the program in multiple runs. For each lipid species, the concentration is normalized to a standard normal distribution (zero mean and unit SD). The partition around medoids (PAM) and K means algorithms were used as base clustering algorithms to run the ConsensusPlus algorithm. The cluster size (K) is set through a range (6 to 13) to explore optimal number of clusters to group the lipids.

Correlation analysis of ceramide concentrations and gene expression

The software for calculating maximal information coefficient was downloaded from <http://www.exploredata.net/> (accessed December 2012), which was maintained by the authors of the report by Reshef et al. (Reshef, Reshef et al. 2011). The statistical significance of the MIC values was determined with the significance table provided by the authors at the threshold of $P < 0.01$. Because the authors only provide the P values for MIC values that are sufficiently large, it is not possible to perform false discovery correction of these P values using the q values (Storey, 2003) package in such setting. The Pearson correlation analysis was performed with the standard R language package. The returned P values for all lipid-versus-gene pairs were further subjected

to false discovery correction with the q value package. The significance threshold is set at $P \leq 0.01$ and $q \leq 0.05$.

4.3 RESULTS

Systematic perturbation of sphingolipid metabolism decouples the biosynthesis of some groups of lipids

Our overall framework of dissecting the functions of specific ceramide species in yeast proceeded as follows: (i) systematically perturb ceramide metabolism using physiological and pharmacological treatments, (ii) monitor lipidomic and transcriptomic responses to the treatments, and (iii) apply systems biology analysis to deconvolute the signaling roles of ceramide species in these responses. Yeast cells were subjected to different combinations (see Materials and Methods for details) of heat stress, ISP1 (myriocin) treatment, and myristate treatment (Figure 4.1A), with each perturbation affecting different part(s) of the lipid metabolic network and leading to diverse lipid profiles. We measured the relative abundance of the ceramide species by mass spectrometry and the changes in gene expression in response to these perturbations using microarrays (Figure 4.1B). We then performed a systems biology analysis to identify correlated changes in ceramide species and gene expression and identified lipid groups that showed similar profiles under all perturbations (Figure 4.1C). Using ontology-based functional analysis and transcription factor analysis (Figure 4.1D and E), we identified functional modules among the genes that were potential targets regulated by a specific ceramide species (or a lipid group). Selected predicted functional associations were validated using phenotypic and transcriptomic experiments (Figure 4.1F).

We first studied ceramide profiles when cells were subjected to heat stress and investigated the impact of blocking de novo synthesis using ISP1, which inhibits the serine palmitoyltransferase (SPT) complex (Figure 9.4), the first committed reaction in the de novo pathway of sphingolipid biosynthesis. Many ceramide species, especially the phytoceramide family (PHC), responded to heat stress through increased de novo synthesis (Figure 4.2A). These included C14, C16, and C18 PHC and α -hydroxy-PHCs (as an example, see inset in Figure 4.2A for C14- α -hydroxy-PHC). In contrast, several members of the dihydroceramide family (DHC) such as saturated C24 and C26 DHC decreased during heat stress in the presence or absence of ISP1 (Figure 4.2A). The decrease of DHCs during heat stress is a novel finding, and the mechanism of how heat stress affects these species has therefore not been defined.

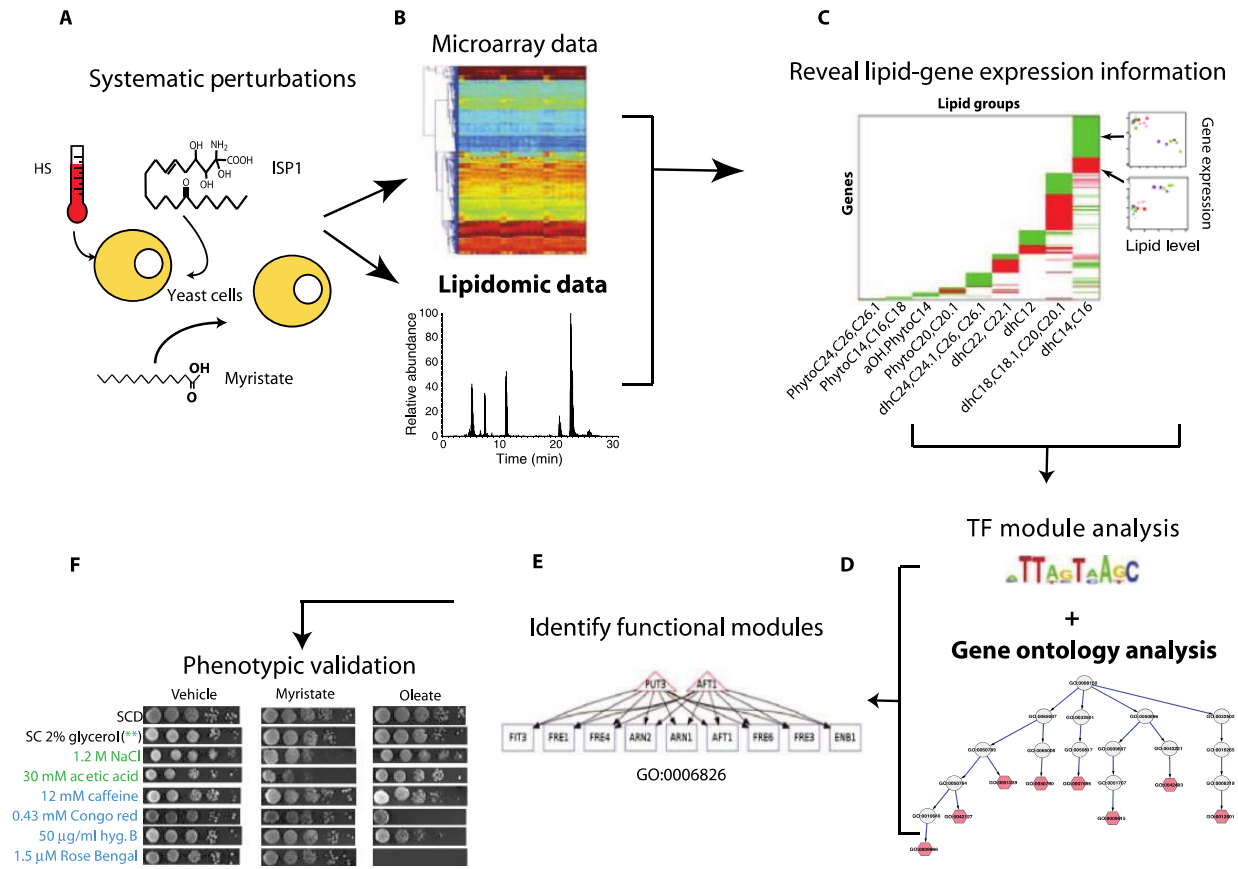


Figure 4.1. Overall strategy of the study. (A) Perturbing sphingolipid metabolism in different experimental conditions: heat stress (HS), ISP1, and myristate treatments. (B) Collecting lipidomic and gene expression (microarray) data. (C) Modeling the relationship between lipids and genes. The pseudocolored matrix shows that different lipid groups (columns) are significantly correlated with different genes (rows). The scatter plots illustrate that genes in the green region of the matrix are negatively correlated with a lipid and that those in the red region are positively correlated with a lipid. (D) Performing ontology-based function analysis and transcription factor (TF) analysis. (E) Identifying functional modules associated with lipid groups. Triangles represent genes encoding transcription factors, rectangles depict genes, and an edge indicates that a gene is regulated by a transcription factor. (F) Validating prediction using phenotypic assay. hyg.B, hygromycinB.

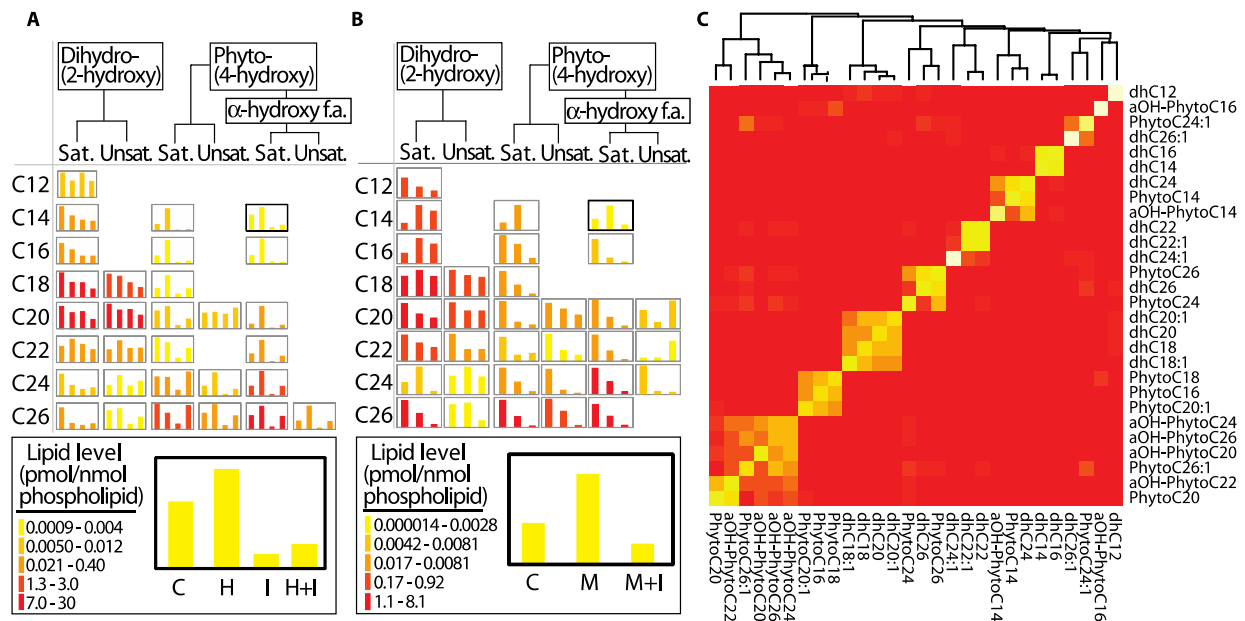


Figure 4.2. Lipidomic analysis. (A) Lipidomic response to combinations of heat stress and ISP1 treatment. Control (C), heat (H), ISP1 (I), and heat plus ISP1 (H+I). (B) Lipidomic response to combinations of myristate and ISP1 treatment. Control (C), myristate (M), and myristate plus ISP1 (M+I). In (A) and (B), rows represent N-acyl chain length, and columns represent single combinations of hydroxyl groups for each ceramide. Saturated (Sat.) and mono-unsaturated (Unsat.) N-acyl chains are indicated. Bar height is averaged triplicate ceramide abundance; the range of each chart is color-coded. Legend inset: C14- α -hydroxy-PHC. (C) Consensus clustering of lipidomic data. The heat map of the consensus matrix reflects how frequently a pair of lipids is assigned to a common cluster during repeated sampling and clustering. A red cell in the matrix indicates that a pair of lipids tends to be assigned to mutually exclusive clusters, and a yellow cell indicates that a pair tends to be assigned to a common cluster. Lipid name abbreviations: dh, dihydro; aOH, α -hydroxy; C followed by a number, fatty acid chain length.

To test the hypothesis that different ceramides regulate distinct cellular signals to mediate cell stress responses, we sought to infer the signaling roles of different ceramide species using gene expression data as readouts of cellular signals. Because of the high connectivity of the sphingolipid metabolic network (Cowart, Shotwell et al. 2010), many species, for example, DHCs differing only in N-acyl chain length, showed correlated changes during heat stress (Figure 4.2A), which obscured potential contributions of individual ceramides or subsets of ceramides. To further dissect and segregate specific ceramide responses, we treated cells with the fatty acid myristate, coupled with treatment with ISP1, to define more specific ceramide responses. Fatty acid treatment changes the concentration of lipid species with a particular fatty acid side chain (Toke and Martin 1996, Al-Feel, DeMar et al. 2003, Petschnigg, Wolinski et al. 2009). Matmati et al. showed that adding different fatty acids with different chain lengths to the medium enriches the PHC pool with those PHC species that correspond to the same chain length (Kim and Tidor 2003). Using this methodology, we treated yeast cells with the long- chain (C14) fatty acid myristate to trigger an acute increase of ceramides with the corresponding C14 acyl chains. Additionally, we also treated the cells with ISP1 to block the incorporation of myristate or palmitate (derived from myristate elongation) into the sphingoid backbone, which would lead to an indiscriminate increase in sphingolipids. Upon myristate treatment, C14, C16, and C24 DHC and C14 PHC species increased (Figure 4.2B). Moreover, several other ceramide species (Figure 4.2B) and the sphingoid bases (C0) decreased in response to myristate, suggesting selective channeling of sphingoid bases to C14 DHC and C14 PHC at the expense of other ceramides. Thus, the C14 and C16 ceramides were effectively decoupled from other ceramides, creating a contrast that would help to resolve the signaling role of these species from other ceramides. To reveal biologically meaningful patterns from the complex lipidomics data sets

collected from the systematic perturbations, we applied consensus clustering analysis (Simpson, Armstrong et al. 2010) to the pooled lipidomic data sets to identify distinct lipid groups. The consensus clustering method repeatedly performs clustering among randomly drawn subsets of the samples to identify intrinsic subgroups of samples, in the current case, the lipids that were inseparable during the repeated clustering. The results showed that ceramides could be further segregated into distinct subgroups (the yellow blocks in Figure 4.2C), identifying lipid subgroups, such as one containing C16, C18, and C20.1 PHCs and one containing C18 and C20 DHCs (Figure 4.2C). Generally, the lipid species that cosegregated into individual ceramide clusters share similar structures and are mostly products of a common set of specific enzymatic reactions in the sphingolipid pathway. For example, the cluster consisting C16, C18, and C20.1 PHCs is separated from the cluster composed of C18, C18.1, C20, and C20.1 DHCs, and synthesis of these ceramide species is metabolically separated by the function of the hydroxylase, Sur2. Clear separation of these clusters indicated that the perturbations induced distinct profiles and decoupled lipids that would exhibit a similar profile if the yeast had only been exposed to a single perturbation, for example, heat stress.

On the basis of the results of clustering analysis and knowledge of ceramide metabolism, we divided the ceramides into nine major groups (Figure 4.4), within which group members were statistically inseparable in the clustering analysis and metabolically inseparable based on biosynthetic pathways. Identification of these clusters lends credence to the theory that enzymes in the sphingolipid metabolism network respond to cellular changes, thus producing distinct profiles for different species. Therefore, we hypothesized that each group functions as a single metabolic, signaling, and functional unit and attempted to identify their corresponding downstream targets using gene expression data and statistical analyses.

Transcriptomic responses are specific to perturbations in sphingolipid metabolism

From the microarray data collected in parallel to the lipidomic data, we identified differentially expressed genes responding to different perturbations (Figure 4.3A). We identified 1893 lipid-mediated stress responding genes that represented the intersections of the heat-sensitive genes with the ISP1-sensitive and with the myristate-sensitive genes. The members of the union gene set were ISP1-sensitive and thus dependent on de novo synthesis of sphingolipids, corroborating previous findings that sphingolipids play an important role in the yeast stress responses (Dickson, Sumanasekera et al. 2006, D. J. Klionsky 2008, Mousley, Tyeryar et al. 2008, Liu, Huang et al. 2012).

To test the hypothesis that distinct ceramides encode disparate signals, which can be detected through the regulation of distinct target gene sets, we studied the relationship between lipidomic and transcriptomic data using three distinct methodologies: (i) the maximum information coefficient (MIC) (Reshef, Reshef et al. 2011), (ii) the Pearson correlation analysis, and (iii) a Bayesian regression model. The MIC quantifies the information between a pair of variables, such as a lipid species profile and a gene expression profile. MIC can capture both linear and nonlinear relationships between variables in a form similar to the familiar correlation coefficient, although the measured association (positively or negatively associated) lacks directionality.

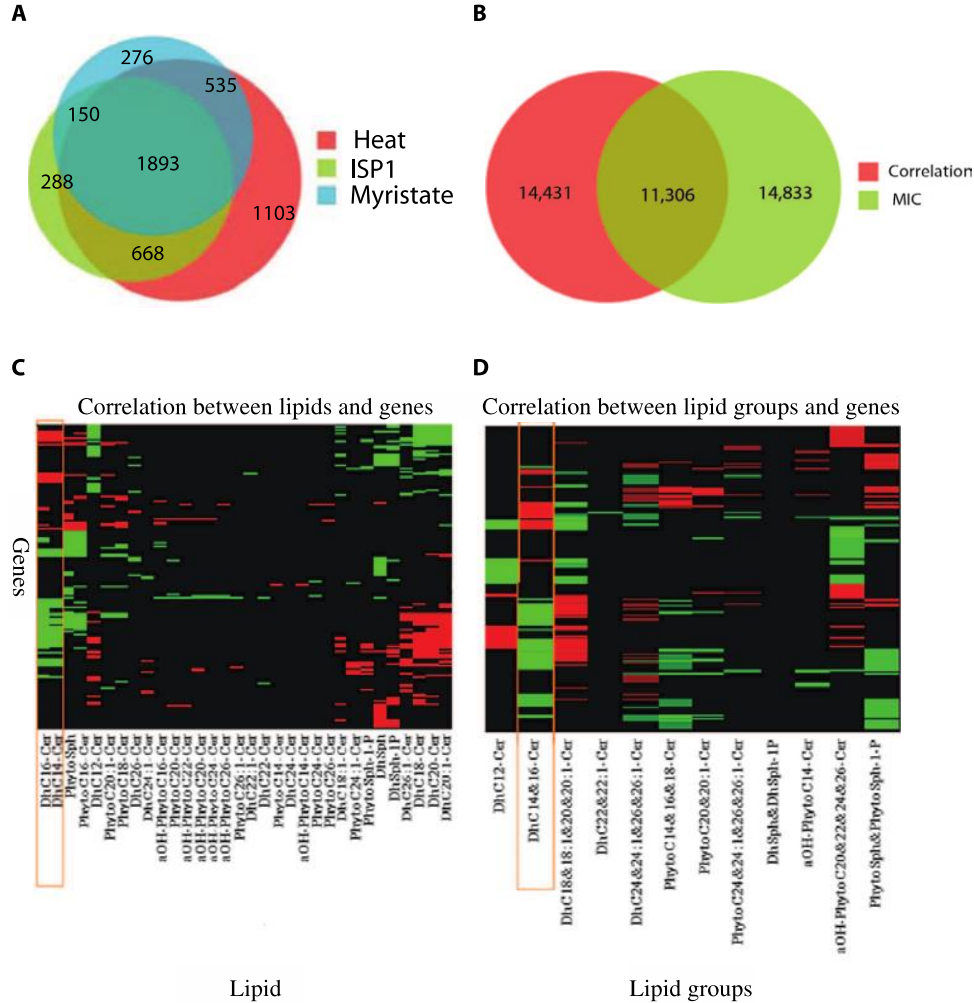


Figure 4.3. Assessing the correlation between lipid abundance and gene expression. (A) Venn diagram illustrating number of genes sensitive to different treatments. (B) Venn diagram illustrating number of lipid-gene pairs with significant association assessed using MIC and Pearson correlation analyses. (C and D) Heat map representation of Pearson correlation coefficient between lipids (C) or lipid groups (D) and gene expression. In the figures, rows correspond to genes that have significant correlation with at least one lipid species, and columns correspond to lipid species. In both figures, a black cell indicates that the corresponding lipid-gene pair is not significantly correlated; a red cell represents that the pair is positively correlated; and a green cell indicates positively that the pair is negatively correlated. Left panel shows the correlation of genes with respect to all lipid species; right panel shows the correlation of genes with respect to lipid groups.

We assessed the significance of MIC and the Pearson correlation of all lipid-versus-gene pairs. A total of 26,139 lipid-gene pairs had significant MIC values ($P < 0.01$; Figure 4.3B); 25,737 lipid-gene pairs had significant Pearson correlation coefficients ($P < 0.01$) with a false discovery threshold q value (Storey, 2003) set at $q < 0.05$. There were non-overlapping portions of the MIC and Pearson sets, which likely reflect the difference in assessing statistical significance between the two methods (Figure 4.3B). We also performed a series of permutation tests in which lipidomic data were randomly permuted to assess the false discovery rate (Good, 1994). None of the Pearson correlation coefficients derived from the permutation experiment passed the threshold of $P < 0.01$ and $q < 0.05$, indicating that the observed relationships between lipids and gene expression were not false discoveries that could result from multiple statistical testing. By Pearson analysis, we identified genes that exhibited a significant correlation, either positive or negative, with at least one lipid species (Figure 4.3C), and clusters of similarly regulated genes were apparent when the correlations were performed on lipid groups (Figure 4.3D). For the third method, we used a regularized regression model (Friedman, Hastie et al. 2010), which represents the expression value (log2-based) of a gene as a linear function of lipids. It progressively shrinks the weighting coefficient of each lipid predictor toward zero if that predictor is not statistically associated with the gene expression, until leaving only a single predictor with a nonzero coefficient. With this model, we achieved the following goals: (i) identifying the most informative ceramide with respect to a gene, (ii) representing the direction of a lipid influence (stimulate or inhibit), and (iii) providing a mathematical means to predict gene expression as a function of lipid concentration (Figure 4.4A). We pooled the genes potentially regulated by each lipid cluster and further grouped them according to the direction of regulation (Figure 4.4A). Each ceramide group had statistically significant parameters with

respect to a set of genes, and the gene sets associated with different ceramides were largely non-overlapping, thus supporting the hypothesis that each species plays roles in distinct pathways regulating different gene sets. We also noticed that distinct gene sets were associated with ceramides with the same head group but different acyl chain lengths, for example, those associated with long-chain (C14 and C16) DHCs were different from those associated with very long chain (C18, C18.1, C20, and C20.1) DHCs (referred to as LC-DHCs and VLC-DHCs, respectively).

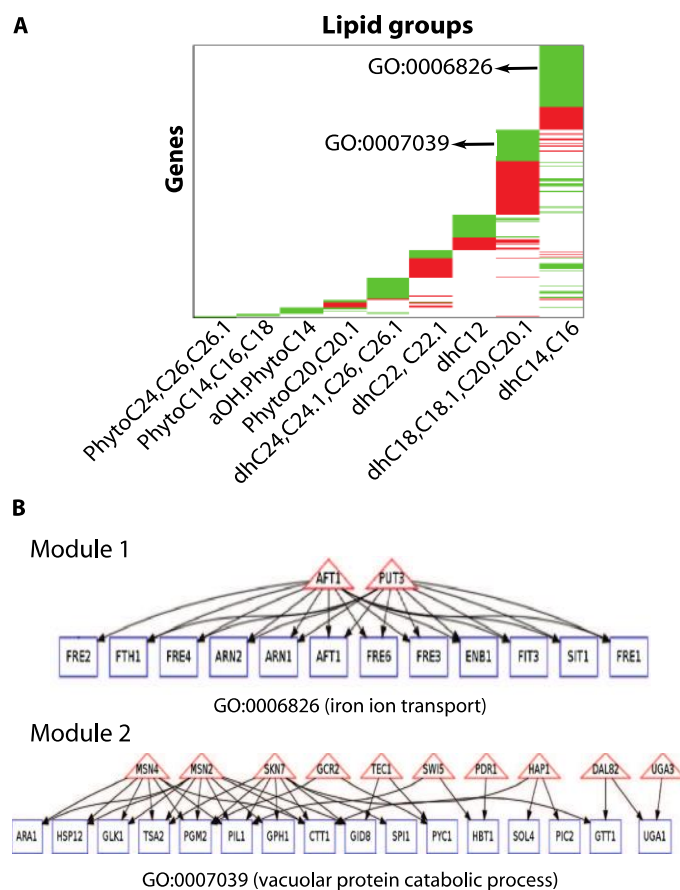


Figure 4.4. Modeling relationship between lipidomic and gene expression data. (A) Organizing genes demonstrating significant correlation with specific ceramides. Genes (rows) are organized according to their association with the different lipid subgroups. A green block represents a set of genes negatively correlated to a lipid, and a red block represents a set of genes positively correlated to a lipid. Examples of major enriched GO terms within gene blocks are shown. (B) Defining pathways of specific biologic modules that respond to specific ceramides, perform related functions, and share transcription factors. Two example modules are shown. Rectangles represent lipid-correlated genes, triangles indicate the transcription factors shared by the genes, and an edge from a transcription factor to a gene indicates that the gene has the binding sites for the transcription factor in its promoter. The function performed by the gens in a module is represented with a GO term.

To better define ceramide-dependent biological processes and to provide mechanistic understanding of ceramide-specific pathways, we performed ontology-based, semantic-driven function analysis and transcription factor analysis of potential target genes. We divided the genes significantly associated with a lipid group into modules (certain genes can be in more than one module) by mining their Gene Ontology (GO) annotations (Chen and Lu 2013), such that each module contains genes that participate in coherently related biological processes, which can be encompassed by a GO term that retains as much of the semantic meaning of their original annotations as possible. Figure 4.4A illustrates that a module of genes involved in the biological process iron ion transportation (GO:0006826) and another module of genes involved in vacuolar protein catabolic process (GO:0007039) were found among the genes negatively correlated to LC-DHCs and VLC-DHCs, respectively. We then applied a graph-based algorithm to search for a set of transcription factors that regulate the members of a module in a cooperative fashion, thus producing a transcription factor module. The analyses, as illustrated by the examples in Figure 4.4B, revealed that the genes in these modules not only performed related functions but also shared transcription factors, which provided mechanistic evidence that the genes in a module were regulated by a common signal. Our functional analyses project the molecular findings from a gene level onto a conceptual level. For example, the results in Figure 4.4B can be translated into the following prediction: “LCDHCs regulate the genes involved in iron ion transportation.” Thus, these gene modules produce testable hypotheses regarding functions of specific groups of ceramide species. All gene modules identified by our analyses—a function map of the ceramide-dependent genes are available at the Web site <http://www.dbmi.pitt.edu/publications/YeastCeramideSignaling>. Note that not all modules have

transcription factors associated with them because of limitations in the available data regarding gene promoters and their regulatory transcription factors.

Heat stress affects DHC metabolism through activation of Ydc1

Heat stress resulted in a decrease in several DHCs through a mechanism that was not inhibited by ISP1 and thus did not require de novo synthesis of sphingolipids (Figure 4.2A). In turn, these changes in DHCs affected the expression of a large number of genes (Figure 4.4A), reflecting their important role in mediating the cellular response to heat stress. Therefore, we investigated the molecular mechanism through which heat stress affected DHC metabolism, more specifically to identify the enzyme(s) that mediates the effect of heat stress. The alkaline dihydroceramidase (encoded by the YDC1 gene) is a good candidate enzyme to mediate the impact of heat stress on long-chain DHCs. Ydc1 hydrolyzes DHCs preferentially over PHCs (Mao, Xu et al. 2000) to a free fatty acid and dihydrosphingosine, thus reducing the concentration of all DHCs. Aft1 is one of the transcription factors associated with the LC-DHC negatively correlated genes, and the AFT1 gene is in the gene module, thus forming a positive feedback loop. Therefore, we analyzed the expression of AFT1 as an indicator of the transcriptional activity of Aft1 in the YDC1 deletion and overexpression yeast strains. We measured AFT1 expression to assess whether Ydc1 was required to mediate changes in gene expression in response to heat stress (Figure 9.5, A and B). Heat stress-induced AFT1 expression (Figure 9.5A) and deletion of YDC1 attenuated the response (Figure 9.5B). Overexpression of YDC1 should decrease DHCs and, thus, mimic the reduction in DHCs caused by the heat stress. Strikingly, overexpression of YDC1 induced AFT1 more than a hundredfold compared with that in wild-type yeast (Figure 9.5C). Thus, our results confirmed that DHCs regulated the expression of AFT1 in module 1, and Aft1 is likely involved in this response. The

results also indicated that activation of Ydc1 is sufficient to induce gene expression changes similar to those induced by heat stress; thus, it is likely one of the enzymes that mediate the impact of heat stress on DHC metabolism.

Phenotypic experiments validate distinct signaling roles of different DHCs

The integrative analyses of lipidomic and transcriptomic data led to the following hypothesis: DHC species with different side chains participate in different signaling pathways. To investigate whether specific transcriptional regulations by distinct DHCs had functional impacts on cells, we examined the effects of perturbing DHCs on cell phenotypes. We focused on the two gene modules shown in Figure 4.2C, which are suggested to be regulated distinctly by LC-DHCs or VLC-DHCs. Because these modules were negatively correlated with the specific DHC groups, we predicted that increasing the respective lipids would repress genes in the corresponding modules and would produce phenotypes mimicking those resulting from deletion of module genes. We identified 17 phenotypes associated with deletion of the genes in the two modules, and we then evaluated yeast cell growth after treatment with myristate or oleate to increase production of the LC-DHCs and VLC-DHCs, respectively.

We analyzed in detail seven phenotypes on the basis of deletion mutant phenotypes (Giaever, Chu et al. 2002, Li, Dean et al. 2002, Motshwene, Karreman et al. 2004, Sambade, Alba et al. 2005, Brombacher, Fischer et al. 2006, Karreman and Lindsey 2007, N. Mira 2010) for the genes within the LC-DHC-sensitive gene module or the VLC-DHC-sensitive gene module (Figure 9.6A). For example, the genes *ARN1*, *ARN2*, and *FRE3* were among iron transport genes that should be negatively regulated by LC-DHCs as predicted by our analysis (Figure 4.4B), and their corresponding deletion mutant strains *arn1D*, *arn2D*, and *fre3D* are all sensitive to high sodium. Increased production of LC-DHCs by myristate treatment, but not

increased VLCDHCs induced by oleate treatment, reproduced this growth defect in the wild-type strain (Figure 9.6B). Conversely, increased production of VLC-DHCs by oleate treatment, and not by myristate treatment, reproduced the Congo red and Rose Bengal sensitivity phenotypes associated with HSP12 and SKN7 deletions (Brombacher, Fischer et al. 2006, Karreman and Lindsey 2007), respectively. We expected that oleate and the corresponding increase in VLC-DHCs would mimic the phenotypes of *hsp12D* because this gene was identified from the microarray data as negatively correlated with VLC-DHCs. *Skn7* is a transcription factor required to induce the genes involved in oxidative responses, and a profound sensitivity of *skn7D* to the singlet oxygen-producing chemical Rose Bengal was reported (Brombacher, Fischer et al. 2006). Our transcription factor analysis indicated that *Skn7* likely stimulates the transcription of seven genes in module 2, thus leading to the hypothesis that VLC-DHCs regulate these genes through suppression of the transcriptional activity of *Skn7*. Oleate treatment led to a marked increase in sensitivity to Rose Bengal in wild-type cells in a lipid-specific manner, which is consistent with the hypothesis that VLCDHCs inhibited the transcriptional activity of *Skn7*. The results from these phenotypic experiments demonstrate the identification of specific functional responses to specific groups of ceramides.

4.4 DISCUSSION

Here, we addressed the challenging task of determining specific signaling roles of distinct ceramides in yeast. In general, a well-established approach to infer causal relationship between two objects (or events) is to manipulate the potential causal object (or event) in a random trial while investigating whether the target object (or event) consistently responds to such

manipulations (C. Glymour 1999). Adopting this principle to lipid-mediated signaling, we applied a series of perturbations to manipulate sphingolipid metabolism, with each leading to unique changes in both ceramide metabolism and gene expression through distinct mechanisms. The results showed significant correlations (linear or nonlinear) between specific ceramide species or ceramide groups and gene expression despite the diversity of lipid and gene response to these perturbations, thus supporting the hypotheses that causal relationships exist between the ceramides and genes studied in this report. Although it is possible that each perturbation may exert effects on gene expression (or phenotypes) through additional confounding mechanisms other than through ceramides, systematic perturbation experiments reduced the likelihood of such effects. For example, the combination of multiple approaches to manipulate LC-DHC—reducing these lipids by heat stress, myrocin treatment, or overexpression of YDC1, and inducing these lipids by myristate treatment—effectively minimizes the impacts of potential confounding factors associated with each individual manipulation. Thus, we confidently concluded that LC-DHCs regulated the genes in module 1. In conclusion, ceramides mediated a multitude of distinct cellular signals in the yeast stress responses. Additionally, this study revealed that the abundance of DHCs was decreased during the yeast response to heat stress, likely through activation of the dihydroceramidase (Ydc1). Functionally, the various DHCs regulated distinct subsets of target genes predicted to participate in distinct biologic processes. Overall, we provided evidence that distinct ceramide species with different N-acyl chains, functional groups, and hydroxylation participate in regulatory processes. The structural complexity of ceramides underscores the potential diversity of the functions that they can play in cellular systems because even closely related ceramides (such as LC-DHCs versus VLC-DHCs) regulated distinct sets of functionally related genes. These findings suggest new research

directions in the study of ceramide-mediated signaling, including their roles in human physiology and disease.

5.0 CHAPTER 2: TRANS-SPECIES LEARNING OF CELLULAR SIGNALING SYSTEMS WITH BIMODAL DEEP BELIEF NETWORKS

5.1 INTRODUCTION

Due to ethical issues, modal organisms such as rat and mouse have been widely used as disease models in studying disease mechanisms and drug actions (Brown 2011, McGonigle and Ruggeri 2014). For example, mouse models have been used to study the disease mechanisms and treatment of type-2 diabetes (Omar, Vikman et al. 2013). Since significant differences exist between species in terms of genome, cellular systems and physiology, the success of using model organisms in biomedical research is hinged on the capability to translate/transfer the knowledge learned from model organisms to humans. For example, when using a rat disease model to screen drugs and investigate the action of drugs, rat cells inevitably exhibit different molecular phenotypes, such as proteomic or transcriptomic responses, when compared with corresponding human cells. Thus, in order to investigate how the drugs act in human cells, it is critical to translate the molecular phenotypes observed in rat cells into corresponding human responses.

Recent species-translation challenges organized by the Systems Biology Verification combined with Industrial Methodology for Process Verification in Research (SBV IMPROVER) (SBV IMPROVER 2013) provided an opportunity for the research community to assess the methods for trans-species learning in systems biology settings (Rhrissorrakrai, Belcastro et al. 2015). One

challenge task was to predict human cells' proteomic responses to distinct stimuli based on the observed proteomic response to the same stimuli in rat cells. More specifically, during the training phase, participants were provided with data that measured the phosphorylation states of a common set of signaling proteins in primary cultured bronchial cells collected from rats and humans treated with distinct stimuli (Poussin 2014). In the testing phase, the proteomic data of rat cells treated with unknown stimuli were provided, and the task is to predict the proteomic responses of human cells treated with the same stimuli (Figure 5.1).

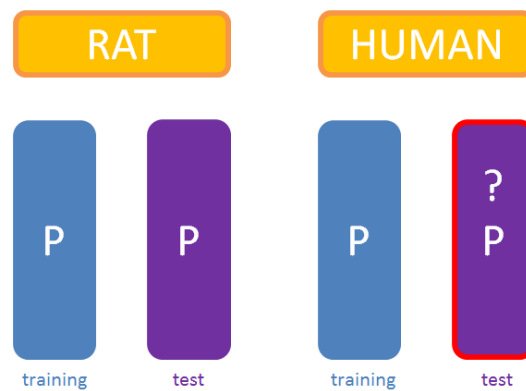


Figure 5.1. Trans-species learning task specification. The objective of the SBV challenge was to predict the phosphorylation states of a set of proteins in human cells treated with different stimuli, based on the observed phosphorylation states of the same set of proteins in rat cells treated with the same stimuli. The blue blocks represent the training data, which are matrices representing the observed phosphorylation states of proteins under different treatment conditions in human and rat cells. The purple blocks represent the test data, in which the phosphorylation states of the set of proteins in rat cells treated with a set of unknown stimuli are provided, and the task is to predict the phosphorylation states of the human cells treated with the same stimuli.

To address the trans-species learning task, a simplistic approach is to train regression/classification models that use the phosphorylation data from rat cells as input features and treat the phosphorylation status of an individual protein from human cells (treated with the same stimulus) as a target class. In this way, predicting the proteomic profile of human cells can be addressed as a series of independent classification tasks or within a multi-label classification framework (Tsoumakas and Katakis 2007, Jin, Muller et al. 2008). However, most contemporary multi-label classification methods treat the target classes as independent or incapable of learning the covariance structure of classes, which apparently does not reflect biological reality. In cellular signaling systems, signaling proteins often form pathways in which the phosphorylation of one protein will affect the phosphorylation state of others in a signaling cascade, and cross-talk between pathways can also lead to coordinated phosphorylation of proteins in distinct pathways (Alberts, Jonson et al. 2008). Another shortcoming of formulating trans-species learning as a conventional classification problem is that contemporary classifiers, such as the support vector machine (Bishop 2006) or regularized regression/classification (Friedman, Hastie et al. 2010), concentrate on deriving mathematical representations that separate the cases, whereas the real goal of trans-species learning is to capture the common signaling mechanisms employed by cells from both model organisms and humans in response to a common stimuli. Indeed, the cornerstone hypothesis underpinning trans-species learning is that there is a common encoding mechanism shared by cells from different species, but distinct signaling molecules are employed by different species to transmit the signals responding to the same environmental stimuli. Therefore, it is important to explore models that are compatible to the above hypothesis. Recent advances in deep hierarchical models, commonly referred to as “deep learning” models (Hinton, Osindero et al. 2006, Hinton and Salakhutdinov 2006, Bengio, Courville et al. 2012),

provide an intriguing opportunity to model the common encoding mechanism of cellular signaling systems. These models represent the signals embedded in observed data using multiple layers of hierarchically organized hidden variables, which can be used to simulate a cellular signaling system because the latter is also organized as a hierarchical network such that signaling proteins at different levels compositionally encode signals with different degrees of complexity. For example, activation of the epidermal growth factor receptor (EGFR) will lead to a broad change of cellular functions including the activation of multiple signaling molecules such as Ras and MAP kinases (Alberts, Jonson et al. 2008), which in turn will activate different transcription factors, eg., Erk-1 and c-Jun/c-Fos complex, with each responsible for the transcription of a subset of genes responding to EGFR treatment. The signals encoded by signaling molecules become increasingly more specific, and they share compositional relationships. Therefore, deep hierarchical models, e.g., the deep belief network (DBN) (Hinton, Osindero et al. 2006), are particularly suitable for modeling cellular signaling systems.

In this paper, we present novel deep hierarchical models based on the DBN model to represent a common encoding system that encodes the cellular response to different stimuli, which was developed after the competition in order to overcome the shortcomings of the conventional classification approaches we employed during competition. We applied the model to the data provided by the SBV IMPROVER challenge and systematically investigated the performance. Our results indicate that, by learning better representations of cellular signaling systems, deep hierarchical models perform significantly better on the task of trans-species learning. More importantly, this study leads to a new direction of using deep networks to model large “omics” data to gain in depth knowledge of cellular signaling systems under physiological and pathological conditions, such as cancer.

5.2 METHODS

In this study, we investigated using the DBN model (Hinton, Osindero et al. 2006) to represent the common encoding system of the signal transduction systems of human and rat bronchial cells. A DBN contains one visible layer and multiple hidden layers (Figure 5.2A). An efficient training algorithm was introduced by (Hinton, Osindero et al. 2006, Hinton and Salakhutdinov 2006), which treats a DBN as a series of restricted Boltzmann machines (RBM) (Figure 5.2B) stacked on top of each other. For example, the visible layer v and the first hidden layer, $h^{(1)}$, can be treated as a RBM, and the first and second hidden layers, $h^{(1)}$ and $h^{(2)}$, form another RBM with $h^{(1)}$ as the “visible” layer. The inference of the hidden node states and learning of model parameters are first performed by learning the RBM stacks bottom up, which is followed by a global optimization of generative parameters using the back-propagation algorithm. In certain cases, edges between visible variables can be added in a RBM to capture the relationship of the visible variables, which leads to a semi-restricted RBM (Figure 5.2C). In the following subsections, we will first introduce the models and their inference algorithms.

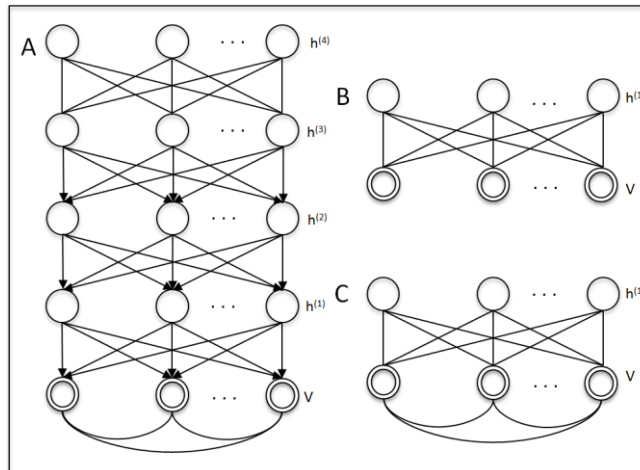


Figure 5.2. Graph representation of the Deep Belief Network and related models. The graph representation of a 4-layered deep belief network. The double circles represent visible variables, and the single circles represent hidden variables. B) The graph representation of a restricted Boltzmann machine. C) The graph representation of a semi-restricted Boltzmann machine in which visible variables are connected.

5.2.1 Restricted Boltzmann Machine (RBMs)

A RBM is an undirected probabilistic graphical model consisting of a layer of stochastic visible binary variables (represented as nodes in the graph) $\mathbf{v} \in \{0,1\}^D$ and a layer of stochastic hidden binary variables $\mathbf{h} \in \{0,1\}^F$. A RBM is a bipartite graph in which each visible node is connected to every hidden node (Figure 5.2B) and vice versa. The statistical structure embedded in the visible variables can be captured by the hidden variables. The RBM model defines the joint distribution of hidden and visible variables using a Boltzmann distribution as follows:

$$Pr(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z(\theta)} \exp(-E(\mathbf{v}, \mathbf{h}; \theta))$$

The energy function E of the state $\{\mathbf{v}, \mathbf{h}\}$ of the RBM is defined as follows:

$$\begin{aligned} E(\mathbf{v}, \mathbf{h}; \theta) &= -\mathbf{a}^T \mathbf{v} - \mathbf{b}^T \mathbf{h} - \mathbf{v}^T \mathbf{W} \mathbf{h} \\ &= -\sum_{i=1}^D a_i v_i - \sum_{j=1}^F b_j h_j - \sum_{i=1}^D \sum_{j=1}^F v_i h_j w_{ij} \end{aligned}$$

where v_i is the binary state of visible variable i ; h_j is the binary state of hidden variable j ; $\theta = \{\mathbf{a}, \mathbf{b}, \mathbf{W}\}$ are the model parameters. a_i represents the bias for visible variable i and b_j represents the bias for hidden variable j . w_{ij} represents the weight between visible variable i and

hidden variable j . The “partition function”, Z , is derived by summing over all possible states of visible and hidden variables:

$$Z(\theta) = \sum_{\mathbf{v}, \mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \theta))$$

The marginal distribution of visible variables is

$$Pr(\mathbf{v}; \theta) = \sum_{\mathbf{h}} Pr(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \theta))$$

5.2.2 Learning Parameter of RBM model

Learning parameters of a RBM model can be achieved by updating the weight matrix and biases using a gradient descend algorithm (delta methods) (Hinton and Salakhutdinov 2006).

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \Delta \mathbf{w}$$

$$\Delta W_{ij} = \epsilon \frac{\partial \log Pr(v)}{\partial W_{ij}} = \epsilon (< v_i h_j >_{data} - < v_i h_j >_{model})$$

where ϵ is the learning rate; $< v_i h_j >_{data}$ is the expected product of the observed data and inferred hidden variables conditioning on observed variables; $< v_i h_j >_{model}$ is the expected product of the model-predicted \mathbf{v} and \mathbf{h} . One approach to derive $< v_i h_j >_{model}$ is to obtain samples of \mathbf{v} and \mathbf{h} from a model-defined distribution using Markov chain Monte Carlo (MCMC) methods and then average the product of the samples, which may take a long time to converge. Representing the $< v_i h_j >_{model}$ derived MCMC chain after convergence as $< v_i h_j >_{\infty}$, one updates the model parameter w_{ij} as follows:

$$\Delta W_{ij} = \epsilon (< v_i h_j >_{data} - < v_i h_j >_{\infty})$$

To calculate $\langle v_i h_j \rangle_\infty$, one can alternatively sample the states of hidden variables given visible variables and then sample the states of visible variables given hidden variables (Salakhutdinov, Mnih et al. 2007) based on the following equations.

$$Pr(h_j = 1|\mathbf{v}) = \sigma(b_j + \sum_{i=1}^n W_{ij}v_i)$$

$$Pr(v_i = 1|\mathbf{h}) = \sigma(a_i + \sum_{j=1}^m W_{ij}h_j)$$

where $\sigma(x)$ is the logistic function $1/(1 + \exp(-x))$.

The convergence of a MCMC chain may take a long time. Thus, to make RBM learning more efficient, we adopted a learning algorithm called contrastive divergence (CD) (Welling and Hinton 2002). Instead of running a MCMC chain for a very large number of steps, CD learning just runs the chain for a small number n of steps and minimizes the divergence between Kullback-Leibler divergence $KL(p_0||p_\infty)$ and $KL(p_n||p_\infty)$ to approximate $\langle v_i h_j \rangle_{model}$ (Carreira-Perpinan and Hinton 2005).

5.2.3 Learning a Deep Belief Network

Unlike a RBM, which captures the statistical structure of data using a single layer of hidden nodes, a DBN strives to capture the statistical structure using multiple layers in a distributed manner, such that each layer captures the structure of different degrees of abstraction. Training a DBN involves learning two sets of parameters: 1) a set of *recognition* weight parameters for the upward propagation of information from the visible layer to the hidden layers, and 2) a set of *generative* weight parameters that can be used to generate data corresponding to the visible layer. The learning of recognition weights can be achieved by treating a DBN as a stack of RBMs and

progressively performing training in a bottom-up fashion (Hinton, Osindero et al. 2006, Hinton and Salakhutdinov 2006). For example, one can treat the visible layer v and the first hidden layer $h^{(1)}$ as a RBM, and then we can treat hidden layer $h^{(1)}$ as a visible layer and form a RBM with the hidden layer $h^{(2)}$. Following the stack-wise learning of RBMs weight parameters and instantiation of hidden variables in the top layer, learning the generative weights across all layers can be achieved by a backpropagation algorithm as in training standard neural networks. The pseudo-code for training a 4-layered DBN is as follows:

Input: Binary data matrix
Output: recognition and generative weights

- 1) Randomly initialize parameters
- 2) Train RBM for layer 1
- 3) Train RBM for layer 2
- 4) Train RBM for layer 3
- 5) Train RBM for layer 4
- 6) Backpropagation

5.2.4 Bimodal DBN

A traditional DBN assumes that data are from one common distribution, and the task is to use distributed hidden layers to capture the structure of this distribution. However, our task of transferring the knowledge learned from rat cells to human cells deviates from the traditional assumption in that humans and rats may use different pathways and signaling molecules to encode the response to a common stimulus. Thus our task is to learn a common encoding system that governs two distributions, which may each have its own mode, hence a bimodal problem. Inspired by the bimodal deep Boltzmann machine model and multimodal deep learning (Ng 2011, Srivastava and Salakhutdinov 2012, Liang, Li et al. 2015), which uses a multi-layered deep network to model the joint distribution of images and associated text, we designed a modified variant of bimodal DBN (bDBN) to capture the joint distribution of rat and human

proteomic data. Our hypothesis is that rat and human cells share a common encoding system that respond to a common stimulus, but utilize different proteins to carry out the response to the stimulus. Thus, we can use the hidden layers to represent the common encoding system, which regulates distinct human protein phosphorylation and rat phosphorylation responses.

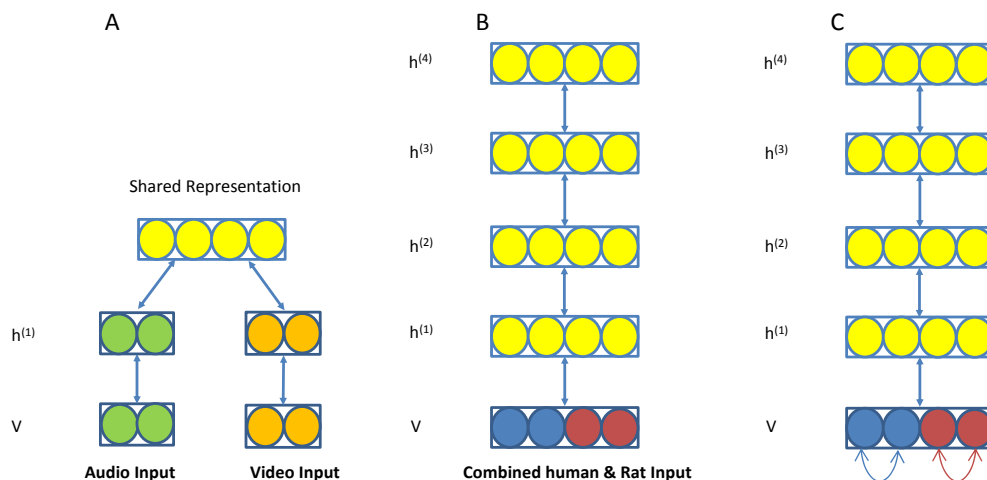


Figure 5.3. Training DBN models. A) A diagram of a conventional bimodal DBN. The green and orange nodes represent different input modalities, e.g., audio and video inputs, and each type is first modeled with a separate hidden layer, and the joint distribution is modeled with a common higher layer hidden nodes. B) A 4-layered bimodal DBN for modeling rat and human proteomic data. The blue and red nodes represent human and rat phosphoproteins respectively. The bottom layer consists of observed variables. Upward arrows represent recognition weights and downward arrows represent generative weights. C) A sbDBN. Additional edges between proteins from the same species are added.

Training

Traditional bimodal models dealing with significantly different input modalities such as audio and video (Figure 5.3A) (Ngiam 2011, Srivastava and Salakhutdinov 2012) usually require one

or more separate hidden layers to first capture the statistical structure of each type of data and then model their joint distribution with common high level hidden layers. However, in our setting, although rat and human proteomic data have their own modalities, they are not drastically different. Therefore, instead of using two separate hidden layers, we devised a modified bimodal DBN, in which a rat training case and a human training case treated with a common stimulus are merged into a joint input vector for the bDBN and connected to a common hidden layer $h^{(1)}$ (Figure 5.3B). In this model, the training procedure is the same as training a conventional DBN using the algorithm described (section 5.2.3), but the prediction is carried out in a bimodal manner. Under this setting, the hidden layers are forced to encode the information that can be used to generate both rat and human data, i.e., the hidden layers behave as a common encoder.

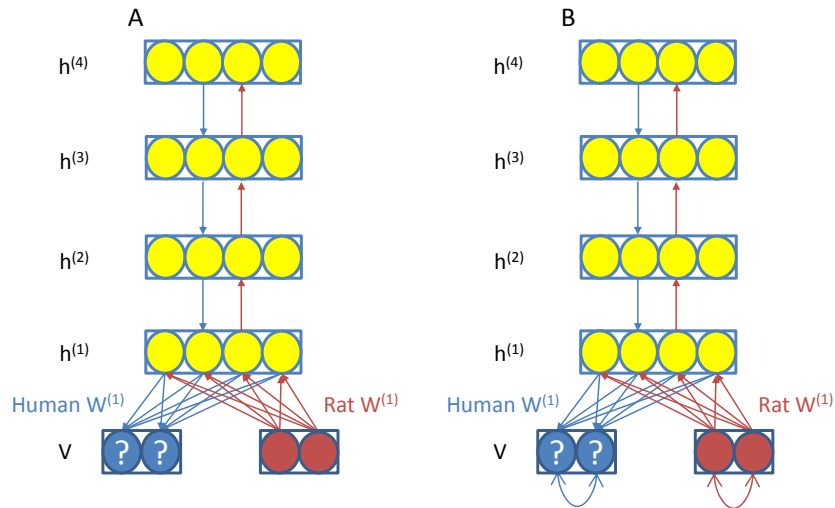


Figure 5.4. Prediction with bDBN and sbDBN models. A) Prediction with bDBN. B) Prediction with sbDBN. When predicting human phosphoprotein states, information derived from rat phosphoprotein states is

propagated upward using weights represented by red arrows and then propagated downwards using the weights represented by blue arrows to predict human phosphoprotein states.

Prediction

When using a trained bDBN to predict human cell response to a specific stimulus based on the observed rat cell response to the same stimulus, we only used the rat data to update the states of nodes in the first hidden layer, $Pr(h^{(1)} | v_{rat})$, with doubled edge weights ($2 \times W^{(1)}_{Rat}$) from rat variables to hidden variables (red edges in Figure 5.4). Then the upper hidden layers were updated using the same method as in a conventional DBN using the recognition weights. When the top hidden layer $h^{(4)}$ was updated using rat data, the bDBN propagated the information derived from rat data downwards to $h^{(1)}$ using generative weights as in a feed forward neural network to predict the human data (Figure 5.4A). We finally predicted the human cell response $Pr(v_{human} | h^{(1)})$ with weights only from hidden variables in $h^{(1)}$ to human visible variables.

5.2.5 Semi-Restricted Bimodal Deep Belief Network (sbDBN)

Since signaling proteins in a phosphorylation cascade have regulatory relationships among themselves, we further modified the bottom Boltzmann machine, consisting of $h^{(1)}$ and v , into a semi-restricted Boltzmann (Taylor and Hinton 2009), in which edges between proteins from a common species are allowed (Figure 5.3C). In this model, the hidden variables in $h^{(1)}$ capture the statistical structure of the “activated regulatory edges” between signaling proteins, instead of “activated protein nodes”. In this model, each human protein was connected to other human proteins, and the same rule was applied to each rat protein. However, we didn’t allow

interactions between human proteins and rat proteins. The interaction between proteins, which is represented as I , was added into the negative phase shown below:

$$\Pr(v_i = 1|h) = \sigma(a_i + \sum_{j=1}^m W_{ij}h_j + I_i)$$

$$I_i = \sum_{k \neq i}^n v_k * \pi_{ik}$$

where I is the influence of the phosphorylation states of other proteins on that of the i th protein.

$$\Delta\pi_{ik} = \epsilon(< v_i v_k >_{\text{data}} - < v_i v_k >_{\text{model}}) \text{ where } k \neq i$$

5.2.6 Performance evaluation

We adopted the evaluation metrics that were used to evaluate and compare the performance of submitted models in the SBV IMPROVER challenge, which include AUROC (area under receiver operator characteristic) (Bradley 1997), AUPRC (area under the precision-recall curve) (Goadrich, Oliphant et al. 2004, Davis and Goadrich 2006), Jaccard Similarity (Dombek, Johnson et al. 2000), Matthews correlation coefficient (Petersen, Brunak et al. 2011), Spearman correlation (Brott, Adams et al. 1989) and Pearson correlation (Adler and Parmryd 2010), to measure the accuracy of the prediction. In all metrics except for Jaccard Similarity, the higher the score, the more accurate the model is. We performed a series of cross-validation experiments, in which we held out the three repeated experiments corresponding to one stimulus of both rat and human cells, performed model training, and test the performance using the held-out samples. All results discussed in the paper were derived from these cross-validation experiments.

5.2.7 Model Selection

When training a deep hierarchical model, often the first task is to determine the structure of the model, i.e., the number of layers and the number of hidden nodes per layer. However, currently there is no well-established method for model selection when training deep learning models. Therefore, we performed a series of cross-validation experiments to search for an “optimal” structure for bimodal and semi-restricted bimodal DBNs. We set the initial structure of both bDBN and sbDBN to the following ranges: $h^{(1)}$: 30 - 50; $h^{(2)}$: 25 - 40; $h^{(3)}$: 20 - 30; and $h^{(4)}$: 20 - 25. We iteratively modified the structure of the model by changing the number of hidden nodes within a layer using a step size of 5 and explored all combinations in the range stated above. In this case, the total number of models tested is 120 ($5*4*3*2$) for both bDBN and sbDBN. Under each particular setting, we performed a leave-one-out experiment to assess the performance of a model. In such an experiment, we held out both human and rat data treated by a common stimulus as the test case, trained models with data treated by the rest of stimuli, and then we predicted the states of human phosphoproteins using the held-out rat data as illustrated in Figure 5.4. By doing this, we predicted human data treated by all stimuli, and we evaluated and compared the performance using the AUROC of different models and retained the model structure that led to the best performance. Note, during leave-one-out training of a model with a given structure, the parameters associated with each model can be different, and therefore the results reflect the fitness of the model with a particular structure after averaging out the impact of individual parameters, an approach closely related to Bayesian model selection (Bishop 2006).

5.2.8 Baseline predictive models

As a comparison to bDBN and sbDBN, we formulated the task of predicting human cell response based on rat cell response to a common stimulus as a classification problem, and we employed two current state-of-the-art classification models, a support vector machine (SVM) (Bishop 2006) with a Gaussian kernel (Karatzoglou, Smola et al. 2004) and an elastic-net regularized generalized linear model (GLMNET) (Friedman, Hastie et al. 2010) to predict human cell responses. In this setting, we trained a classification model (SVM or GLMNET) for one human protein using a vector of rat proteomic data collected under a specific condition as input features (independent variables) and the human protein response under the same condition as a binary class variable (dependent variables). We trained one such classifier for each human protein class. We performed leave-one-out cross-validation using SVM and GLMNET models respectively. The results predicted by SVM and GLMNET were then compared with the results predicted by DBN and sbDBN.

5.3 RESULTS

5.3.1 The Data

The protein phosphorylation response data in this study was provided by SBV IMPROVER (SBV IMPROVER 2013). The data contains the phosphorylation status of 16 proteins collected after exposing rat and human cells to 26 different stimuli (Table 1). Each stimulus was repeated 3 times. The SBV IMPROVER organizers preprocessed the proteomic data into binary values to

represent if a protein was phosphorylated under a specific condition. We directly utilized the binary input for our DBN models.

Table 1. Proteins and stimuli involved in this study

Stimuli	5AZA, AMPHIREGULIN, BETAHISTINE, BISACODYL, CHOLESTEROL, CLENBUTEROL, EGF, EGF8, FLAST, FORSKOLIN, HIGHGLU, IFNG, IGFII, IL4, MEPTYRAMINE, NORETHINDRONE, ODN2006, PDGFB, PMA, PROKINECITIN2, PROMETHAZINE, SEROTONIN, SHH, TGFA, TNFA, WISP3, DME
Proteins	AKT1, CREB1, FAK1, GSK3B, HSPB1, IKBA, KS6A1, KS6B1, MK03, MK09, MK14K11, MP2K1, MP2K6, PTN11, TF65, WNK1

5.3.2 Model selection results

In order to identify the “optimal” model structure that perform well, we examined the performance of each model with a specific structure configuration stated in Section 2.7. For a given model, we performed a leave-one-out cross validation experiment and calculated the AUROC for the model. The average of the AUROCs for 120 bDBN models was 0.80, and the highest one is 0.86. The bDBN structure yielding the best AUROC consisted of four hidden layers with the following numbers of nodes 35, 30, 30 and 20, from $h^{(1)}$ to $h^{(4)}$ respectively. For sbDBN, the mean of the AUROCs for 120 candidate models is 0.86 and the highest one was 0.93. The number of nodes for the four layers for the best sbDBN model was 30, 30, 30 and 20, from $h^{(1)}$ to $h^{(4)}$ respectively. A tentative explanation for the different numbers in $h^{(1)}$ between bDBN and sbDBN is that the edges between the visible variables in the sbDBN partially captured the statistical structures of the visible variables, which reduced the need for additional

nodes in the layer $h^{(1)}$. In the following sections, we report the results derived from bDBN and sbDBN with these two specific structures with the highest AUROCs.

Hyper Parameters used for model training

The weights were updated using a learning rate of 0.1, momentum of 0.9 and a weight decay of 0.0002. The weights were initialized with random values sampled from a standard normal distribution multiplied by 0.1. Contrastive divergence learning was started with $n=1$ and increased in small steps during training.

5.3.3 Comparison among different models

Table 2 shows the comparisons between different predictive models in terms of 6 evaluation metrics. We highlighted the best value for each metric using bold face letters. When comparing bDBN with SVN and GLMNET, the results show that bDBN performs better in term of AUROC and Spearman's correlation, but underperformed in terms of AUPRC, Jaccard similarity, and Pearson correlation. This is potentially due to the fact that we performed model selection mainly using AUROC as the criteria. Strikingly, with the addition of protein-protein edges in the visible layer, the 4-layered sbDBN performs much better than all other models measured in all metrics.

Based on the AUROC value, the performance of the 4-layered sbDBN > 4-layered bDBN > SVM > GLMNET. However, ranking varies depending on the scoring method. It is known that models pursuing optimal area under the ROC curve is not guaranteed to optimize the area under the Precision-Recall curve (Davis and Goadrich 2006). Indeed, we noted that the AUROC for the 4-layered DBN is better than the one for GLMNET. However, the AUPRC for the 4-layered DBN is worse than the one for GLMNET (Table 2).

Table 2. Leave-one-out accuracy scores of models

	AUPRC	AUROC	Jaccard. Similarity	Matthews. Correlation. Coefficient	Speaman. Correlation	Pearson. Correlation
4-layered bDBN	0.417	0.859	0.750	0.373	0.323	0.235
4-layered sbDBN	0.632	0.936	0.531	0.616	0.391	0.460
SVM	0.493	0.724	0.692	0.411	0.231	0.392
GLMNET	0.444	0.709	0.717	0.374	0.194	0.282

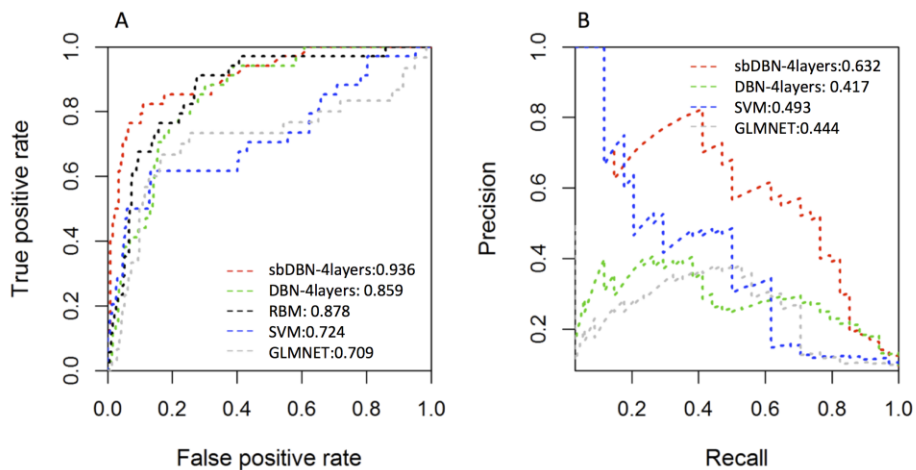


Figure 5.5. ROC and RPC curves of different models. A) Performance results of four models in terms of AUROC. B) Performance results of four models in terms of AUPRC.

5.3.4 Biological interpretation of learned edges between proteins in sbDBN

The best predictive power of the sbDBN reflects the importance of capturing the correlation between signaling proteins. We then investigated whether the learned correlations between signaling proteins are biologically sensible, although it should be noted that Boltzmann machine

models cannot infer causal relationships. For each protein, we picked the top 3 strongest interaction edges for rat and human respectively, and we organized the results as shown in Figure 5.6. In this figure, if the interaction between a pair of proteins exists in both rat and human data, the edge is colored green. If the interaction is rat only, there is a blue line between the two proteins. If the interaction is human only, there is a red line between the two proteins. The results indicate that, while some common correlations are shared between rat and human cells, different covariance structure exists in different proteomic data.

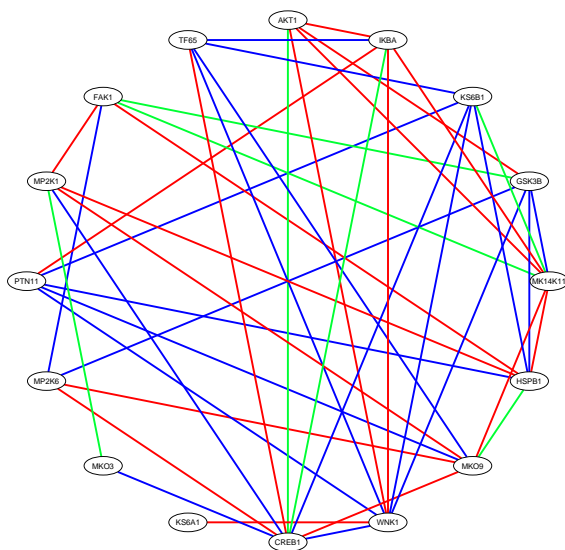


Figure 5.6. Protein correlation network learned from the 4-layered sbDBN. Ovals represent the proteins. A green line represents a common edge shared between human proteins and rat proteins; a red line represents an edge between human proteins; a blue line represents an edge between rat proteins.

Due to the fact that signal transduction in live cells are dynamic events, it is difficult to thoroughly evaluate the accuracy of inferred interactions even through further experimentations.

Conventional evaluation metrics such as sensitivity and specificity are difficult to assess in this study. Since it is possible that the signal transduction between a pair of proteins known to have a regulatory relationship may not be present under the experimental conditions of this study, accurately assessing sensitivity is challenging; similarly, since there are seldom reports or databases stating that signal transduction never occurred between a pair of proteins, it is challenging to assess if the lack of an edge between a pair of proteins in our model really represents a true negative outcome. As such, conventional metrics such as AUROC cannot be applied in our evaluation. However, we noted that we were able to assess with reasonable confidence the positive predictive value (PPV) of the model, i.e., the percentage of the predicted signal transduction interactions that is known in literature. We performed a comprehensive literature review and cited the references supporting the predicted regulatory relationship and known protein-protein interactions in Supplementary Tables. The results indicate that most of the predicted regulatory relationships are supported by the literature or have evidence of physical interactions between the proteins. Thus, the results support the notion that the sbDBN correctly captured the correlation (thereby signal transduction or cross talks) between phosphoproteins.

5.4 DISCUSSION

In this study, we investigated the utility of novel deep hierarchical models in a trans-species learning setting. To our knowledge, this is the first report using deep hierarchical models to address this type of problem. Our results indicate that, by learning to represent a common encoding system for both rat and human cells, the deep learning models outperform contemporary state-of-the-art classifiers in this trans-species learning task.

The empirical success of deep hierarchical models may be attributed to the following advantages. First, the DBN is capable of learning novel representations of the data that are salient to the task at hand. The DBN models are more compatible to the biological systems that generate the observed data. The hidden variables at the different layers of the DBN models can capture information with different degrees of abstraction, thus allowing the models to capture a more complex covariance structure of the observed variables. It is possible that hidden nodes at lower layers, e.g., $h^{(1)}$, directly capture the covariance of the observed protein phosphorylation states, whereas the higher layers can capture the crosstalk between signaling pathways that only occur in response to specific stimuli. Thus, shallow models that only concentrate on the covariance at the level of observed variables, such as SVM and elastic network, would have difficulties capturing such a high-level covariance structure of the data. It is now well appreciated that feature-learning methods, such as DBN, tend to outperform feature selection methods in complex domains, such as image classification and speech recognition (Hinton, Osindero et al. 2006, Hinton and Salakhutdinov 2006, Bengio, Courville et al. 2012). Second, DBN strives to learn the common encoding system for both human and rat data, and it naturally performs multi-label classification by taking into account the covariance of the class variables. However, a conventional classifier, such as an SVM, can only predict one human protein as the class variable in an independent manner, thus failing to capture the covariance of class variables and yielding inferior performance.

The sbDBN model developed in this study provides a novel approach capable of simultaneously learning interactions and predicting the state of phosphoproteins. Interestingly, the model assigns differential weights to the edges between phosphoproteins when comparing those from rat and human cells, which potentially indicates that different parts of signaling

pathways are preferentially utilized in a species-specific manner. However, this hypothesis still needs to be experimentally tested in a relatively larger dataset.

Deep hierarchical models are particularly suited for modeling cellular signaling systems, because signaling molecules in cells are organized as a hierarchical network and information in the system is compositionally encoded. Our results indicate that DBNs were capable of capturing the complex information embedded in proteomic data. Interestingly, in contrast to the training of deep learning models in a machine learning setting such as object recognition in image analysis where usually a large number of training cases is required, our results show that the DBN models performed very well given a moderate size of training cases. This indicates that biological data tend to have strong signals that can be captured by DBNs with relative ease. Our study demonstrates the feasibility of using deep hierarchical models to simulate cellular signaling systems in general, and we foresee that deep hierarchical models will be widely used in systems biology. For example, one can use deep hierarchical models to study how cells encode the signals regulating gene expression, to detect which signaling pathway is perturbed in a specific pathological condition, e.g., cancer. Finally, models like our bDBN and sbDBN provide a novel approach to simultaneously model multiple types of “omics” data in an “integromics” fashion.

6.0 CHAPTER 3: LEARNING A HIERARCHICAL REPRESENTATION OF THE YEAST TRANSCRIPTOMIC MACHINERY USING AN DEEP AUTOENCODER MODEL

6.1 INTRODUCTION

A cell constantly responds to its changing environment and intracellular homeostasis. This is achieved by a signal transduction system that detects the signals, assimilates the information of diverse signals, and finally transmits its own signals to orchestra cellular responses. Many of such cellular responses involve tightly regulated transcriptomic activities, which can be measured by microarray or RNA-seq technology and used as readouts reflecting the state of the cellular signaling system.

Reverse engineering the signaling system controlling gene expression has been a focus area of bioinformatics and systems biology. However, this task is significantly hindered by the following difficulties: 1) a transcriptomic profile of a cell (with contemporary technology, often a population of cells) at a given time represents a convolution of all active signaling pathways regulating transcription in the cells, and 2) the states of the majority of these signaling pathway are not observed, making it a challenging task to infer which genes are regulated by a common signal pathway, and it is even more challenging to reveal the relationships among signaling pathways.

Different latent variable models, such as principle component analysis (Raychaudhuri, Stuart et al. 2000), independent component analysis (Liebermeister 2002), Bayesian vector quantizer model (Lu, Hauskrecht et al. 2004), network component analysis (Liao, Boscolo et al. 2003, Devarajan 2008), and non-negative matrix factorization (Brunet, Tamayo et al. 2004, Devarajan 2008) models have been applied to analyze transcriptomic data, with an aim to represent the states of latent pathways using latent variables. Despite the different strengths and limitations of these models, they share a common drawback: the latent variables in these models are assumed to be independent, i.e., the latent variables are organized in single “flat” layer without any connection among them; as such the models lack the capability of representing the hierarchical organization of cellular signaling system.

Figure 6.1 illustrates the task of reverse-engineering a transcriptomic regulation system. Figure 6.1A illustrates the well-appreciated hierarchical organization of signaling molecules in cells and how the information encoded by signaling molecules are compositionally organized. It also shows that the convoluted signals eventually are emitted as changed gene expression. At this stage, all the hierarchical information of the signaling system is embedded in the data, a vector of gene expression value, in the form of context-specific and compositional covariance structures. When given a collection of transcriptomic profiles collected under different cellular conditions (Figure 6.1B), the ultimate task is to recover the structure of the signaling systems shown in Figure 6.1A, but the goal remains unattainable with current methodologies. In this study, we hypothesize that the hierarchical organization of cellular signals can be partially reconstructed by models capable of discovering and representing the context-specific and compositional covariance structure embedded in transcriptomic data. To this end, recent development in deep hierarchical models, commonly referred to as “deep learning” models, e.g.,

the autoencoder (deep belief network) shown in Figure 6.1C, afford us the tools to reverse engineer the signaling systems of cells by mining systematic perturbation data.

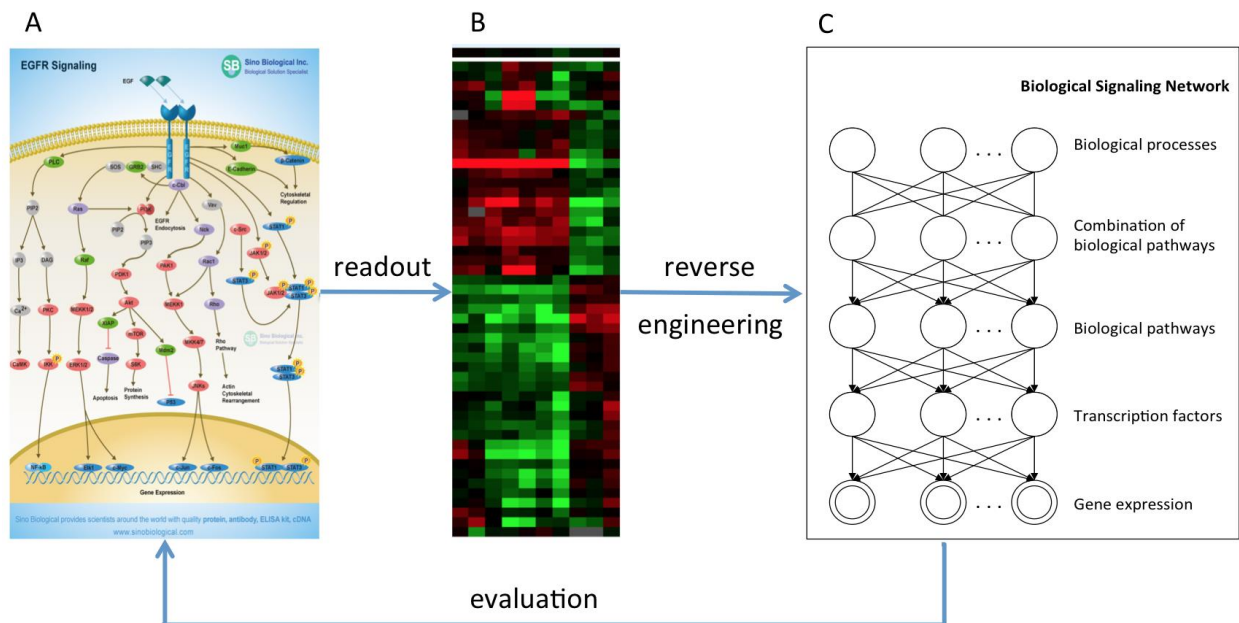


Figure 6.1. Overview of studying molecular signaling transduction using an autoencoder. (A) An example of molecular signaling transduction. (B) An example of the heatmap of gene expression microarrays. (C) An autoencoder model consisting of hierarchically organized hidden variables. After the model was trained, we evaluated the information learnt from the autoencoder model by testing whether the information carried by hidden variables in the autoencoder has real biological entities.

In this family of deep hierarchical models, multiple layers of hidden (latent) variables are organized as a hierarchy, which can be used to capture the compositional relationships embedded in the transcriptomic data in a distributed fashion, i.e., different layers can capture different degrees of detail. For example, the relationships between TFs and their target genes can be

captured by a hidden variable layer (hereafter referred to as hidden layer) immediately above the observed layer of observed gene expression variables, whereas the function of pathways regulating TFs can be represented by higher hidden layers. Therefore, deep hierarchical models provide an abstract representation of the statistical structure of the transcriptomic data with flexibility and different degrees of granularity. We hypothesize that, if accurately trained, a deep hierarchical model can potentially represent the information of real biological entities and further reveal the relationships among them.

In this study, we designed and trained a sparse deep autoencoder model to learn how the information is encoded in yeast cells when subjected to diverse perturbations. Our results indicate that deep learning models can reveal biologically sensible information, thus learning a better representation of the transcriptomic machinery of yeast, and we believe that the approach is applicable to more complex organisms.

6.2 METHODS

In this study, we investigated using the autoencoder model (Hinton, Osindero et al. 2006) and sparse autoencoder model (Lee 2008) to represent the encoding system of the signal transduction systems of yeast cells. Before introducing the autoencoder model and sparse autoencoder model, we will first briefly review restricted Boltzmann machines (RBMs) as building blocks for the autoencoder.

6.2.1 Restricted Boltzmann Machines (RBMs)

A RBM is an undirected probabilistic graphical model that consists of two layers of stochastic binary variables (represented as nodes in the graph): a visible layer $v \in \{0,1\}^D$ and a hidden layer $h \in \{0,1\}^F$. The energy function E of the state $\{v, h\}$ of the RBM is:

$$E(v, h; \theta) = -a^\top v - b^\top h - v^\top W h = -\sum_{i=1}^D a_i v_i - \sum_{j=1}^F b_j h_j - \sum_{i=1}^D \sum_{j=1}^F v_i h_j w_{ij}$$

In this equation, the binary state of visible variable i is represented by v_i , the binary state of hidden variable j is by h_j and the model parameters are $\theta = \{a, b, W\}$. The bias for visible variable i is a_i , the bias for hidden variable j is b_j and the weight between visible variable i and hidden variable j is w_{ij} .

The joint distribution of the hidden and visible variables is defined using a Boltzmann distribution, and the conditional probability of the states of hidden variables and visible variables are as follows:

$$Pr(v, h; \theta) = \frac{1}{Z(\theta)} \exp(-E(v, h; \theta))$$

$$Z(\theta) = \sum_{v, h} \exp(-E(v, h; \theta))$$

$$Pr(h_j = 1 | v) = \sigma(b_j + \sum_{i=1}^m w_{ij} v_i)$$

$$Pr(v_i = 1 | h) = \sigma(a_i + \sum_{j=1}^n w_{ij} h_j)$$

where $\sigma(x)$ is the logistic function $1/(1 + \exp(-x))$, m is the total number of visible variables and n is the total number of hidden variables.

The efficient algorithm for learning parameters of the RBM model was introduced in detail in literature and our previous work (Hinton, Osindero et al. 2006, Hinton and Salakhutdinov 2006, Chen, Cai et al. 2015)

6.2.2 Autoencoder

Unlike a RBM, which captures the statistical structure of data using a single layer of hidden nodes, an autoencoder uses multiple layers in a distributed manner, such that each layer captures the structure of different degrees of abstraction. As shown in Figure 6.1C, an autoencoder contains one visible (input) layer and one or more hidden layers. To efficiently train the autoencoder, we treat it as a series of two-layered restricted Boltzmann machines (RBM) stacked on top of each other (Hinton, Osindero et al. 2006, Hinton and Salakhutdinov 2006). The inference of the hidden node states and learning of model parameters are performed by learning the RBM stacks bottom-up, which is followed by a global optimization of generative parameters using the back-propagation algorithm. More details of the algorithm and pseudo code for training an autoencoder were discussed in both literature and our previous work (Hinton, Osindero et al. 2006, Hinton and Salakhutdinov 2006, Chen, Cai et al. 2015).

6.2.3 Sparse autoencoder

In a conventional RBM model, each hidden unit is fully connected to the observed variables. After training, there is usually a non-zero weight between each pair of visible and hidden nodes. Based on the assumption that the change in gene expression due to a specific perturbation—most microarrays in this study are experiment-vs-control—is likely mediated by a small number of

TFs or pathways, we adopted the sparse autoencoder model (Lee 2008, Goh 2010) to simulate the cellular response to perturbations. The sparse autoencoder model enables one to specify that only a certain percent of hidden nodes have a high probability to be set to 1 (“on”) by adding a penalization term to the optimization function. Optimization of the traditional RBM is performed by minimizing the negative log-likelihood of the data during RBM training within an autoencoder:

$$\text{minimize}_{\{\theta\}} - \sum_{l=1}^s \log \sum_{j=1}^n Pr(v^l, h_j^l | \theta)$$

where s is the total number of samples, n is the total number of hidden units and $\theta = \{a, b, W\}$.

The sparse RBM adds the regularization term (Lee 2008) into the optimization:

$$\text{minimize}_{\{\theta\}} - \sum_{l=1}^s \log \sum_{j=1}^n Pr(v^l, h_j^l | \theta) + \lambda \sum_{j=1}^n |p - \frac{1}{s} \sum_{l=1}^s E[h_j^l | v^l]|^2$$

where λ is the regularization constant and p is a constant (usually representing the percent of nodes desired to be on) controlling the sparseness of the hidden units h_j . For the traditional RBM, the parameters are updated just based on the gradient of the log-likelihood term. But for the sparse RBM, the parameters are updated not only based on the gradient of the log-likelihood term but also the gradient of the regularization term. As mentioned in the RBM section (Appendix A.3), the update of parameters for a RBM without the addition of the regularization term is as follows:

$$\Delta w_{ij} = \epsilon(< v_i^+ h_j^+ > - < v_i^- h_j^- >)$$

$$\Delta b_j = \epsilon(< h_j^+ > - < h_j^- >)$$

$$\Delta c_i = \epsilon(< v_i^+ > - < v_i^- >)$$

The form of update changes when the regularization term is added. The regularization term only penalizes the activation of the hidden units. So only the update of the weights and the biases of hidden units is changed; the update of the biases of visible units is still the same.

$$\begin{aligned}\Delta w_{ij} &= \epsilon(\langle v_i^+ h_j^+ \rangle - \langle v_i^- h_j^- \rangle) - \lambda \langle v_i^+ (h_j^+ - \rho_j) \rangle \\ \Delta b_j &= \epsilon(\langle h_j^+ \rangle - \langle h_j^- \rangle) - \lambda \langle (h_j^+ - \rho_j) \rangle \\ \Delta c_i &= \epsilon(\langle v_i^+ \rangle - \langle v_i^- \rangle)\end{aligned}$$

6.2.4 Non-negative matrix factorization

Non-negative matrix factorization (NMF) has been applied to reduce the dimension of expression data from thousands of genes to a handful of hidden representations (ex. metagenes) (Brunet, Tamayo et al. 2004). NMF is an algorithm based on decomposition by parts that can reduce the dimension of a matrix V (Devarajan 2008).

$$V = W * H$$

Given that the gene expression data is represented as matrix V , NMF factorizes it into a basis matrix (W) and a coefficient matrix (H). All three matrices should have non-negative elements. The number of hidden regulators is pre-defined, and is usually much smaller than the number of genes. In this study, we used the Matlab function nonnegative matrix factorization “nnmf” to perform NMF analysis.

6.2.5 Model selection of autoencoder and sparse autoencoder

We performed a series of cross-validation experiments to search for an “optimal” structure for autoencoders and sparse autoencoders. We adopted a four-layered autoencoder to represent the

hierarchical structure of biological processes shown in Figure 6.1C. We then explored models with different numbers of hidden units in each hidden layer. We set the initial structure of both autoencoder and sparse autoencoder to the following ranges: $h^{(1)}$: 100 - 428; $h^{(2)}$: 50 - 100; $h^{(3)}$: 50; and $h^{(4)}$: 25. We iteratively modified the structure of the model by changing the number of hidden nodes within a layer using a step size of 50 for the first and second hidden layer. Then we explored all combinations in the range stated above. In this case, the total number of models tested is 14 (7*2) for both autoencoder and sparse autoencoder. For the sparse autoencoder, we chose three sparsity constants that are 0.05, 0.1 and 0.15. Under each particular setting, we performed ten-fold cross-validation to assess the performance of a model.

We used two criteria of evaluating the performance of the models. One is the reconstruction error, which is the difference between the original input data and the reconstructed data after training the model (Vincent 2008). Due to the sparse features of the sparse autoencoder, we used Bayesian information criterion (BIC) (Posada and Buckley 2004) as another criteria for comparing models. BIC combines the factors of likelihood and number of free parameters to be estimated. The model with the lowest BIC is preferred.

$$\text{BIC} = -2 \cdot \ln \hat{L} + k \cdot \ln(n)$$

$$\hat{L} = \prod_{i=1}^N p^m (1-p)^{1-m}$$

where \hat{L} is the maximized value of the likelihood function of the model, k is the number of free parameters to be estimated, n is the number of samples, p is the probability predicted from the model for a gene to be active in an experiment, and m is the true binary state of a unit in the input data. More details of model selection could be found at section B.3.

6.2.6 Mapping between the hidden units and known biological components

Based on the weights between each hidden unit in the first hidden layer and all the visible units (genes), we used a threshold (top 15% of the absolute values of weights) to cut the edges between a hidden node and the observed genes, such that an edge indicates that the hidden node regulates the gene. We then identified all genes predicted to be regulated by a hidden node as a gene set. Based on the DNA-Protein interaction table (Huang and Fraenkel 2009, Yeger-Lotem, Riva et al. 2009), we also identified the gene set regulated by a known TF. We then assessed the significance of overlapping of gene sets regulated by hidden nodes and TFs using hypergeometric testing.

We have two gene sets that are assigned to *geneset1* and *geneset2*. Both gene sets are represented by a binary vector with 1 to be regulatory genes and 0 to be non-regulatory genes. We wanted to use the enrichment analysis to see whether the regulatory genes in *geneset1* are similar to the regulatory genes in *geneset2* with pre-defined functions. With these two datasets ready, we could calculate the enrichment score between the two gene sets. We ran the enrichment analysis using hypergeometric testing. The R function for the hypergeometric testing is as follows:

$$dhyper(x, m, n, k)$$

where x represents the number of intersected regulated genes between *geneset1* and *geneset2*, m represents the number of regulated genes in *geneset1*, n represents the number of un-regulated genes in *geneset1*, and k represents the number of regulated genes in *geneset2*. We used the p -value of the enrichment score as a criterion for whether two gene sets are similar.

6.2.7 Consensus clustering of experiment samples

Consensus cluster clustering (Monti 2003) was used to cluster the experiment samples using different datasets as input. The R implementation of ClusterCons (Simpson, Armstrong et al. 2010) was downloaded from CRAN (<http://cran.r-project.org/web/packages/clusterCons/>). The inputs for consensus clustering are the samples represented using original gene expression values, NMF megagenes values and the states of hidden variables under all experiment samples respectively. The partition around medoids (PAM) and *K*-means algorithms were used as base clustering algorithms. The inputs for cluster by cluster consensus clustering are the samples represented using samples clusters derived from the nodes from different hidden layers as features. If one sample belongs to a sample cluster, its input value is 1. Otherwise, its input value is 0.

6.2.8 Finding pheromone related hidden units

We calculated the significance of the mapping between the state of a hidden node and the state of proteins related to the pheromone signaling pathway by using the chi-square test. Figure 6.2 lists the pheromone related proteins from our published paper (Lu, Jin et al. 2013). First, we used a threshold (top 15%) to designate the state of a hidden unit as active or inactive based on its activation probability (Table 3). Then, for each hidden unit, we created a contingency table to collect the counts of the joint state of the hidden node and whether any member of the pheromone pathway is perturbed in a specific experiment. We used the contingency table (Table 4) to perform the chi-square test, and used a *p*-value of 0.01 as the significance threshold.

6.2.9 Gene ontology analysis

GO (Ashburner, Ball et al. 2000) provides a standard description of what gene products do and where they are located. One of the frequently used databases that provide GO information for yeast is Saccharomyces Genome Database SGD. We first used the combination of weights (Dumitru 2010) between neighboring hidden layers to get the weights between the hidden units in a particular hidden layer and the genes. A gene is regarded as being regulated by a hidden unit if their weight is in the top 15% of all weights. When a gene set of interest associated with a hidden unit is available, we used the method mentioned in (Chen and Lu 2013) to summarize the GO terms capturing as much as semantic information associated with those genes (Lu and Lu 2012). We identified the GO terms that could summarize the largest number of genes, while undergoing a minimal information loss.

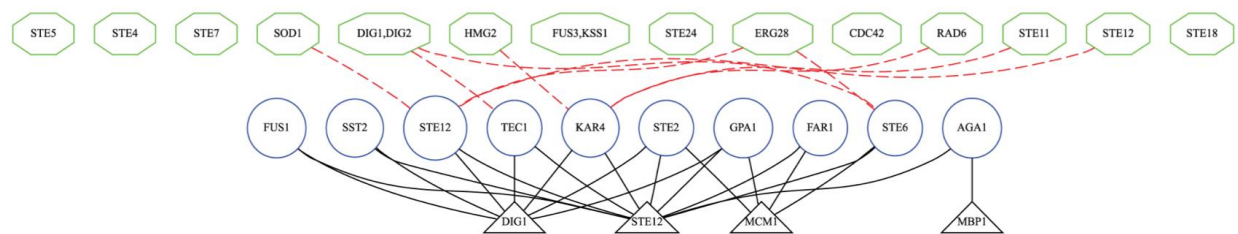


Figure 6.2. Pheromone signaling pathway related proteins.

Table 3. Binary values of pheromone-related samples and states of hidden units

Samples	State of proteins related to pheromone signaling pathway	State of a hidden unit
Pheromone unrelated	0	0
Pheromone related	1	0
...
Pheromone related	1	1

Table 4. Contingency table between the states of samples and the states of each hidden unit

Chi-square test	Hidden-unrelated	Hidden-related
Pheromone-unrelated	Count (0, 0)	Count (0, 1)
Pheromone-related	Count (1, 0)	Count (1, 1)

6.2.10 Incorporation of Prior Knowledge into the DBN model

The motivations of incorporating prior knowledge into DBN models are as follows:

1) With the generation of input data reflecting prior knowledge, model could be enforced to learn the right information. The power of reconstruction should be improved. 2) The addition of simulated data increases the number of samples, which reduces the probability of over-fitting.

Most of the models with incorporation of knowledge know what variables stand for (ex. Bayesian network), which allows for the addition of prior knowledge into the model directly. However, in our DBN model, all the nodes are latent. This makes it really hard to incorporate the prior knowledge into model directly. One of the solutions is to generate input data reflecting prior knowledge and treat it as real input data. In our study, the prior knowledge is the interaction between TFs and genes.

Here we simulate gene expression input data by incorporating prior knowledge of TF-gene interactions. For each TF, we produce a binary vector according to the protein-gene interactions. The values of genes regulated by a TF are 1 and the rest are 0. We add some noise into the generated data by adding a flip rate such as 0.01 (only 1% of the values are flipped to the opposite). For each TF, we produce 5 samples of gene expression data accordingly. The flip rate could be changed to reflect the effect of prior knowledge. For example, if the flip rate is set near to 1, then the data is almost randomly generated and doesn't reflect any prior knowledge. Since the flip rate would be one of the model parameters, we will need to find a flip rate that results in a relatively small reconstruction error. The number of TFs in the database is 214. Therefore, by using the procedures above, we will simulate 1070 gene expression data. Then we will merge the simulated gene expression data with our original 1690 gene expression data to train the DBN model.

For evaluation, we use a t-test to test whether the model that incorporates prior knowledge is better than the one without it. We leave some samples out of the original data to be used as test data and use the rest to train the model. When the model is trained, we use the test data as input to calculate the regenerated test data. The reconstruction error is calculated based on the regenerated test data and the original test data. Then we use a t-test to compare the reconstruction error obtained by the model with prior knowledge and the model without it, and see whether these two models are significantly different from each other. If the p-value of the t-test is smaller than the threshold 0.05, then it shows that the two models are significantly different from each other. If not, it shows that the original model is good enough without prior knowledge.

6.2.11 Identification and visualization of information represented by latent variables in higher hidden layers

Similar with the visualization of latent variables in the task of image recognition, which uses input pixels strongly associated with a hidden unit (H Lee 2008), we identified signals (signal proteins/signaling pathways) represented by a hidden unit by finding input genes that are strongly associated with it. In this study, we applied weight linear combination (2.3.5) to get the weight matrix between the hidden layer of interest and the visible input layer. The genes strongly associated with a hidden unit (top 5% of weight values) are then analyzed further by performing GO analysis (6.2.9) and gene set enrichment analysis to evaluate the conceptual functions they perform.

6.3 RESULTS AND DISCUSSION

6.3.1 Training different models for representing yeast transcriptomic machinery

We collected a compendium of 1,609 yeast cDNA microarrays from the Princeton University Microarray Database (puma.princeton.edu), and we combined them with 300 microarrays from the study by Hughes et al. (Hughes, Marton et al. 2000), which was used in a previous study of the yeast signaling system (Lu, Jin et al. 2013). The combined dataset is ideal for studying the yeast signaling system because it represents a large collection of perturbation experiments that are of biological interest. For example, the data from the study by Hughes et al (Hughes, Marton et al. 2000) were collected from yeast cells with genetic perturbations (deletion of genes) or

chemical treatments, and similarly the microarrays from the database were collected from specific conditions and contrasted with “normal” growth condition. Taking advantage of the experiment-vs-control design of cDNA microarrays, we identified differentially expressed genes (3-fold change) in each array and retained 2,228 genes that were changed in at least 5% of the microarrays. We then represented the results of each microarray experiment as a binary vector, in which an element represented a gene, and its value was set to 1 if the gene was differentially expressed, and 0 otherwise. Thus, each microarray represented the *transcriptomic changes* in response to a certain condition, presumably regulated by certain specific signaling components, which is unknown to us.

We investigated the utility of the autoencoder model (also known as deep belief network) (Hinton, Osindero et al. 2006), with one observed layer representing the microarray results and 4 hidden variable layers (hereafter referred to as hidden layers) representing the yeast signaling components in yeast transcriptomic machinery. In this model, a hidden node is a binary variable, which may reflect the state of a collection of signaling molecules or a pathway, such that the switching of the node state between 1 and 0 can reflect the changing state of a pathway.

The probabilistic distribution of the state of a node in a given layer is dependent on the nodes in the adjacent parent layers, defined by a logistic function. The directed edges between nodes of adjacent layers indicate that, conditioning on the state of nodes in parent layer, the nodes in a child layer are independent. In other words, the statistical structure (patterns of joint probability of nodes) among the nodes in a child layer is captured by the nodes in the parent layer. For example, in our case, if the nodes in the 1st hidden layer (directly above the gene expression layer) represent the states of transcription factors, then co-differential expression (covariance) of a set of genes is solely dependent on (or explained by) the TFs that regulate the

genes. Similarly, the co-regulation of TFs is determined by its parent layer, which may reflect the state of signaling pathways. Thus, this model is suited to capture the context-specific changes and compositional relationship among signaling components in a distributed manner. The model is referred to as autoencoder because, when given a collection of observed data, it learns to use hidden nodes to encode the statistical structure of observed data, and it is capable of probabilistically reconstructing the observed data.

Since the autoencoder model in our study is biologically motivated, we hypothesize that the nodes in the first hidden layer would likely capture the signal of TFs. Thus the number of nodes in this layer should be close to the number of known TFs for yeast, of which there are around 200 well-characterized yeast TFs (Harbison, Gordon et al. 2004). However, for a given microarray from a perturbation experiment, genes that respond to a specific perturbation are likely regulated by a few transcription factors. Thus we also investigated a model referred to as *sparse autoencoder* (Lee 2008, Ng 2011), which performs regularized learning of model parameters and allows a user to constrain the percent of nodes in a layer that can be set to the “on” state, see Methods for details. In our experiment, we constrained that, in the first hidden layer, around 10% of hidden nodes should be used to encode the changes in a microarray.

We first evaluated how adding a sparse regularization term influenced the state of hidden units (the probability of hidden units to be active/on). We trained a conventional autoencoder and a sparse autoencoder (setting the sparsity constraint to 10%) using the microarrays. For each microarray, the models probabilistically inferred the state of each hidden node (the probability of a node to take a value of 1). Figure 6.3 shows the histogram of the expected states of the nodes in the first hidden layer associated with all microarray samples. In the conventional autoencoder, a relatively larger number of the nodes in the first hidden layer had a non-zero probability to be 1

(“on state”) (Figure 6.3A), whereas the majority of the hidden nodes in the sparse autoencoder model were expected to take a value of 0 (“off state”) (Figure 6.3B). Thus, the sparse autoencoder strives to use less hidden nodes to encode the same statistical structure in the observed data, instead of using every hidden node, with each contributing a little to the expression of genes. This is a desired property conforming to our assumption that the response to a specific perturbation should be encoded by a relatively small number of TFs.

We further evaluated how well models with different architectures (mainly concentrating on the number of hidden nodes in the first hidden layer) can be used to represent the transcriptomic machinery of yeast. Table 12 shows the results of a limited model selection experiment based on our biological assumptions (note that an exhaustive search of possible combinations of architecture and parameter settings is intractable). We calculated the reconstruction error for different architectures (Table 11), which is the sum of the differences across all microarrays between the observed expression state (0 or 1) of genes in microarrays and the expected states of genes reconstructed by the DBN. Table 5 lists the reconstruction errors for DBN and sparse DBN. The student’s t-test shows that the models with different architectures are significantly different from each other ($p\text{-value} < 0.05$). However, the difference of reconstruction error between architectures of the same type of model (ex. autoencoder) is small compared with the difference between autoencoder and sparse autoencoder. The results indicate that models of the same type (conventional or sparse autoencoder) could learn to encode data with a similar accuracy across the range of the architectures studied here, although the sparse autoencoder had higher reconstruction errors.

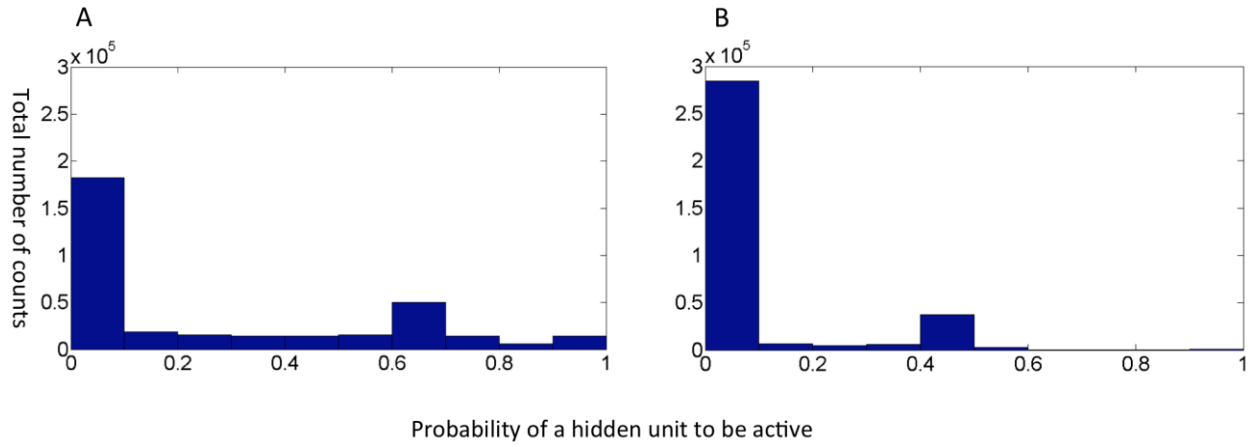


Figure 6.3. Histogram of the expected states of hidden units (probability of hidden units to be on) in the first hidden layer for the conventional autoencoder (A) and sparse autoencoder (B) respectively. For both models, the number of hidden units from the first hidden layer to the fourth hidden layer is 214, 100, 50, and 25 respectively. The sparsity threshold for the sparse autoencoder is 0.1. A hidden unit has a state under each experiment condition. Therefore, the total number of states for all hidden units is the number of experiment condition (1609) * the number of hidden units (214). The x-axis is the probability of a hidden unit to be on ranging from 0 to 1. The y-axis is the count of states.

Table 5. Reconstruction error of models with different architectures

Reconstruction error	Architecture 1 (100:100:50:25)	Architecture 2 (150:100:50:25)	Architecture 3 (214:100:50:25)	Architecture 4 (428:100:50:25)
autoencoder training	150.20	150.23	148.94	150.81
autoencoder test	188.57	189.12	190.76	189.63
Sparse autoencoder training (0.1)	170.19	170.07	170.41	171.94
Sparse autoencoder test (0.1)	206.58	208.40	203.99	203.27

Table 6. BIC scores of different models

	Arch 1 (100,100,50,25)	Arch 2 (214, 100, 50, 25)
Autoencoder	3.25e+006 = 1.99e+006 + 1.26e+006;	4.41e+006 = 1.71e+006 + 2.70e+006;
Sparse autoencoder	2.06e+006 = 1.93e+006 + 1.26e+005;	1.96e+006 = 1.69e+006 + 2.70e+005;

The cells show the BIC score (bold) and the individual terms of the BIC (see Methods). The numbers in the parentheses associated with each architecture (Arch) indicate the number of hidden nodes in 1st – 4th hidden layers.

While the results indicate that the reconstruction errors of sparse autoencoder models were a bit higher than the ones of the traditional autoencoder, it should be noted that the sparse autoencoder reconstructed the same data with a much smaller number of hidden variables. From the perspective of the minimum description length (MDL) principle (Barron 1998), a model is preferred if it can encode the information of a dataset with a minimal description length while achieving a similar or better reconstruction of data. In information theory, the description length is measured as the number of bits needed to encode the data, and in our case each bit is encoded by a hidden node. Thus, the sparse autoencoder potentially is a more desirable model even if it suffers a higher reconstruction error. To quantify and compare the utility of conventional and sparse autoencoders, we calculated the Bayesian information criteria (BIC) of the models, and the results are shown in Table 6. The results indicate that the BIC of the sparse autoencoder with an architecture consisting of hidden layers with 214, 100, 50, 25 hidden nodes (1st to 4th)

respectively is the lowest (the best) among the compared models. Since the number of hidden nodes in the first hidden layer of this model agrees better with the knowledge of the number of transcription factors, we chose to investigate the results derived from this model in the following sections.

6.3.2 Distributed representation enhances discovery of signals of TFs

The motivation of using a hierarchical model is to allow latent variables in different hidden layers to capture information with different degree of abstraction in a distributed manner. When modeling transcriptomic data, one goal is to discover the signals of TFs. In a sparse autoencoder, it is natural to expect that the 1st hidden layer should capture the signals encoded by TFs. We test this hypothesis by evaluating the overlap of the genes predicted to be regulated by a hidden node in the 1st hidden layer and those known to be regulated by a TF. A statistically significant overlapping between them is shown in Figure 6.4.

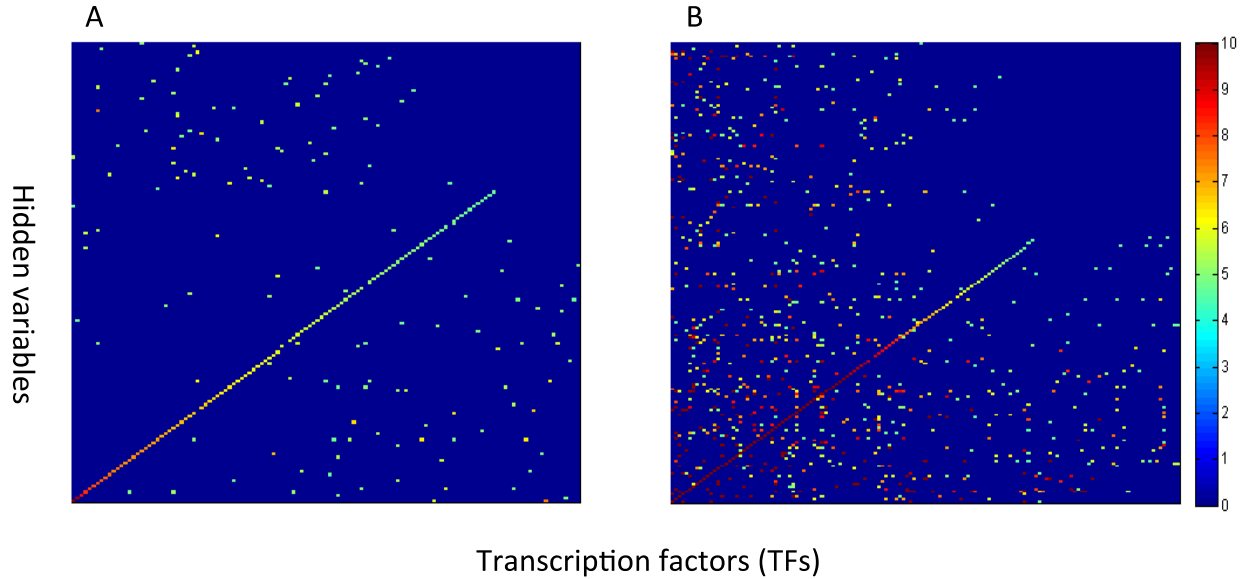


Figure 6.4. Mapping between transcription factors (TFs) and hidden variables in the first hidden layer. Results for sparse autoencoder (A) and NMF (B) are shown. The transcription factors (TFs) are arranged along x-axis, and the hidden variables are arranged along y-axis. Each point in the figure represents the value of $-\log(p\text{-value})$ of the enrichment score between genes regulated by a hidden node and a TF. The pseudo-color bar shows the scale of the $-\log(p\text{-value})$.

Indeed, the results indicate that sparse autoencoder is capable of capturing and representing the information of TFs, in that there is an almost one-to-one mapping between hidden nodes and known TFs as shown in Figure 6.4. For a few hidden variables that are significantly mapped to multiple TFs, we further investigated if these TFs are members of known TF complexes (Hill, Marais et al. 1993). As an example, we found that a hidden node is significantly mapped to *AFT1* and *PUT3*, which are two yeast TFs known to cooperatively regulate genes involved in ion transportation (Alkim, Benbadis et al. 2013). As another example, a hidden node is mapped to both *MSN2* and *MSN4* (Fabrizio, Pozza et al. 2001), which belong to the same family and form hetero-dimers to regulate genes involved in general stress response.

To demonstrate the advantage of hierarchical and distributed representations, we compare our results with another latent variable model commonly used to represent microarray data, the non-negative matrix factorization model (NMF) (Devarajan 2008). The NMF can be thought of as a model consisting of an observed layer (gene expression) and a single hidden layer (hidden variables/metagenes (Brunet, Tamayo et al. 2004)), which is used to capture all signals in embedded in microarrays, whereas the same information is distributed to multiple layers of hidden variables in the sparse autoencoder. We trained a NMF model with 214 “metagenes”, which is the same as the number of hidden nodes in the 1st hidden layer of the sparse autoencoder, and the results of mapping between latent factors and TFs are shown in Figure 6.4.

Table 7 shows the quantitative comparisons between the four-layered sparse DBN and non-hierarchical NMF. The number of TFs mapped to hidden units by the sparse DBN is more than the number of TF mapped to hidden units by NMF. Besides, for the sparse DBN, the average number of TFs mapped to a hidden unit is 4. For NMF, the average number of TFs mapped to a hidden unit is 22. Even though the relatively large number of TFs mapped to a hidden unit using NMF may be due to the existence of TF complexes or strong interactions among TFs, there is still other complex information involved which is hard to interpret. For the sparse DBN, the complex information, such as TF complex, could be captured by the second hidden layer. When we compare the genes associated with the hidden units in sparse DBN and NMF, we found that the gene set associated with some hidden units in NMF are enriched in the gene set associated with some hidden units in the third hidden layer of the sparse DBN. This shows that the sparse DBN could use different hidden layers to capture information of different complexity. The first hidden layer mainly captures specific information about TFs. The higher hidden layer captures complex information, such as TF complexes and signaling pathways.

Table 7. Quantitative comparisons between the four-layered sparse DBN and non-hierarchical NMF.

Parameter Settings	Number of sig-enrich TF-node pairs	Number of hidden variables	Number of TFs	Average enrichment p-value (-log)
Sparse 214,100,50, 25-0.15	242	158	144	5.547938
Nonnegative Matrix Factorization (0.05)	934	176	127	9.406635

Indeed, the results clearly demonstrate the expected difference between the two models. With the capability of capturing the context-specific and compositional relationship of signals regulating expression in a distributed manner, the hidden nodes in the 1st hidden layer clearly capture the specific signals of TFs, whereas the signals regulated TFs are delegated to the higher level hidden nodes. In contrast, with only a single layer of latent variables, all signals in the data need to be “squeezed” into these latent variables, such that a latent factor (a “metagene”) has to represent the signal of multiple TFs. Therefore, the results support our hypothesis that, benefitting from the distributed representation of the statistical structures across multiple hidden layers, the sparse encoder can concisely learn and represent the information of biological entities, in this case the TFs.

6.3.3 Latent variables can capture the information of signaling pathways

We further investigated whether certain hidden nodes can represent the states of well-known yeast signaling pathways, i.e., whether the state of a hidden node can be mapped to the state of a collection of proteins in a pathway. In a previous study (Lu, Jin et al. 2013), we were able to recover the pheromone signaling pathway and a set of target genes whose transcription were

regulated by the pheromone pathway, by mining the systematic perturbation data from the study by Hughes et al (Hughes, Marton et al. 2000) in which 14 genes involved in yeast pheromone signaling were perturbed by gene deletion. In the current study, we identified the microarray experiments in which the aforementioned 14 pheromone-pathway genes were perturbed, and we examined if the state of any hidden node was statistically associated with perturbation of pheromone pathway, using the chi-square test (see Methods). Interestingly, we found 2 hidden nodes in the 1st hidden layer that are significantly associated with the perturbation experiments, one with a chi-square test $p \leq 2.47\text{e-}05$, and the other with a $p \leq 3.82\text{e-}02$. Further inspecting the genes predicted to be regulated by these hidden nodes, we found a significant overlap between the pheromone target genes from our previous study and the genes regulated by these hidden nodes (data not shown). These results indicate that the hidden nodes of the sparse autoencoder model are capable of capturing the signals of specific yeast pathways. However, it should be noted that, by design, a hidden node in the high level layers of the sparse autoencoder might encode the signals of multiple pathways that share strong covariance.

6.3.4 The hierarchical structure captures signals of different degrees of abstraction

One advantage of the hierarchical structure of an autoencoder is to represent information with different degrees of abstraction. Intuitively, the lower level hidden layers should capture the highly specific signaling pathways or signaling molecules, such as TFs, whereas the high level hidden layers may encode more general information. To test this hypothesis, we identified the genes regulated by each hidden node by performing a linear weight combination experiment (Dumitru 2010) (multiplication of weights between different hidden layers). We then applied a semantic analysis method previously developed by our group (Chen and Lu 2013), which

identifies the most appropriate concept from the Gene Ontology (GO) to summarize the genes. Section 6.3.2 showed the mapping between hidden units in the first hidden layer and TFs. For the second, third and fourth hidden layer, the average number of significant GO terms (with p-value < 0.05) associated with each hidden unit is 15, 21 and 25 respectively. The higher the hidden layer, the larger the number of genes associated with a hidden unit is. Therefore, it is reasonable that the number of significant GO terms associated with a hidden unit in a higher hidden layer is more than the ones associated with a hidden unit in a lower hidden layer. Interestingly, we found that genes regulated by AFT1 and PUT3 are significantly enriched among the genes regulated by a hidden node in the 2nd hidden layer, and the genes regulated by this hidden node is summarized by the GO term GO:0006826 (*iron ion transportation*), shown in Figure 6.5. Using the same method, we found another hidden node whose related genes were annotated with GO:0006357 (*regulation of transcription from RNA polymerase II promoter*). As we mentioned above, there could be multiple GO terms associated with a hidden node. The terms GO:0006826 and GO:0006357 discussed above are the two with the most specific biological function and the most significant p-value. Figure 6.6 shows an example of the top 3 GO terms associated with a hidden unit. Besides the GO terms, the hidden units in the higher hidden layer could also be represented by KEGG pathways. Figure 6.7 is an example of the representation of a hidden unit in the second hidden layer related to pheromone signaling pathway. Ko04011 and GO0071444 were found to be the most strongly associated with that node using KEGG pathway analysis and GO analysis separately. The other yeast pathway databases, such as REACTOME, could also be used to represent hidden units. The results indicate that the distributed representation of information enables the hidden nodes at the different levels of hierarchy to capture information of different degrees of abstraction.

Table 8. Two hidden units found to be the most related to pheromone signaling

	Index of hidden variable	p-value	-log(p-value)
The best score	160	2.47e-05	10.60659
The 2nd score	128	3.817e-02	3.2654

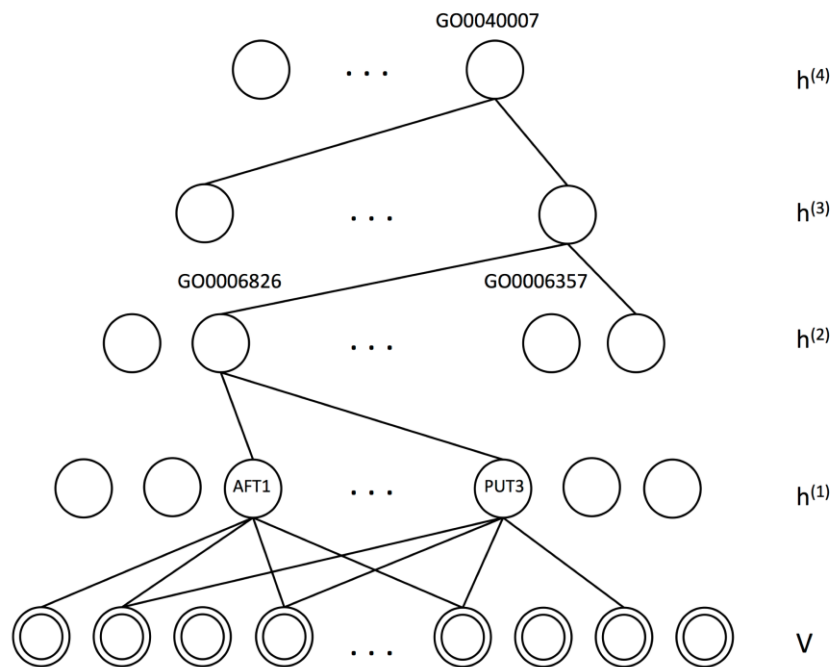


Figure 6.5. Example of hierarchical Gene Ontology (GO) map for hierarchical hidden structure.

GO term	GO term Name	Number of genes annotated	Enrichment p-value(-log)
GO0006826	Iron ion transport	30	9
GO0097286	Iron ion import	24	6
GO0034756	Intracellular protein transport	15	5

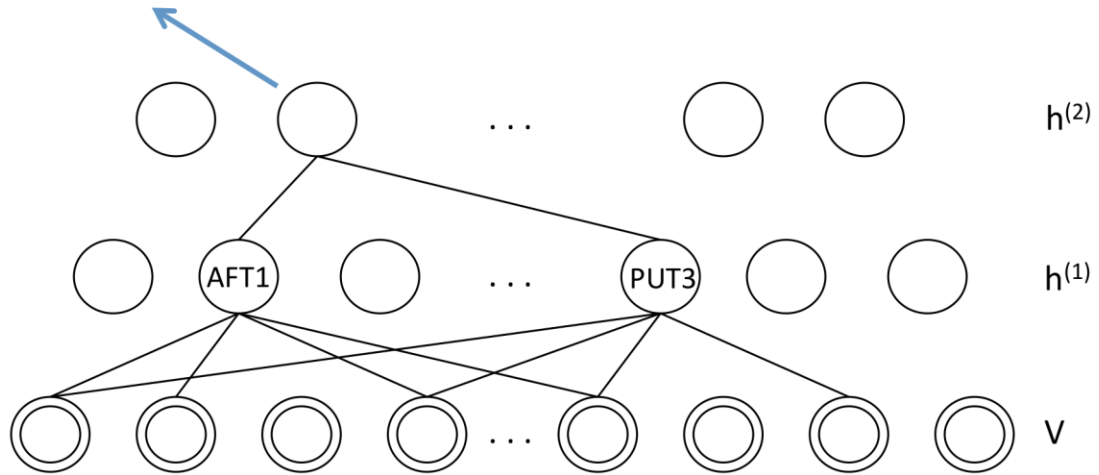


Figure 6.6. Example of multiple GO terms associated with a hidden unit in a higher hidden layer.

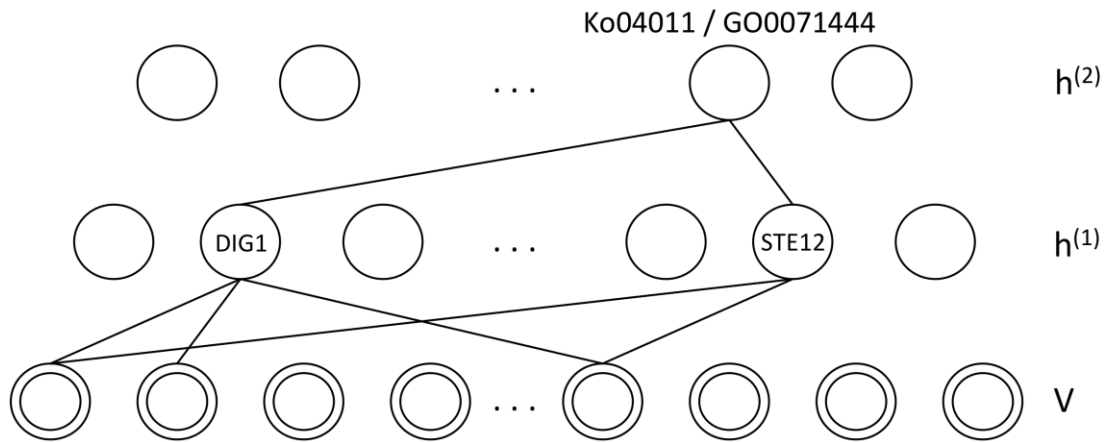


Figure 6.7. Example of the representation of a hidden unit related to pheromone signaling pathway.

The hidden unit could either be represented by a GO term or a KEGG pathway. Ko04011: MAPK signaling pathway. GO0071444: cellular response to pheromone. DIG1 and STE12 are two TFs in the first hidden layer, which are strongly associated with the node GO0071444 in the second hidden layer.

6.3.5 Concise representation enhances the discovery of global patterns

Given a large comprehensive dataset, it is often interesting to learn if distinct perturbations affect common biological processes of the cell (Hughes, Marton et al. 2000, Lu, Jin et al. 2013). A common approach to discover such patterns is to perform clustering analysis and examine if certain samples (thereby experimental perturbations) are clustered together. In general, the result of a clustering analysis is significantly influenced by whether the features representing each sample are informative. Non-informative features may not reveal any real information, whereas features concisely reflecting the states of cellular signaling system may provide insights regarding the system. To examine if the signals represented by the latent variables are more informative than original gene expression values and NMF metagene values, we represented the samples in our dataset using the original gene expression values, NMF metagene values and the

expected states of the hidden nodes in a hidden layer respectively, and we then compared the results of consensus clustering (Figure 6.8).

The results clearly demonstrate that, if samples are represented in the high-dimensional gene expression space, the majority of the samples cannot be grouped into distinct clusters. When we use the low-dimensional NMF metagenes space, it performs slightly better than using original gene expression space. But, the clusters derived are still not well separated. In contrast, when samples were represented using the expected states of hidden nodes of the 1st hidden layer, the samples can be consistently separated into distinct clusters. Representing samples using the expected states of the nodes from other hidden layers also produced clearly separated clusters (data not shown). The results indicate that the states of hidden nodes are much more informative in terms of representing distinct characteristics of individual samples, thus enabling clean separation of samples by the clustering algorithms. Interestingly, we found that some pheromone signaling related mutations, such as STE5, STE26, FUS3 and RAD6, were included in one sample cluster in Figure 6.8C. Most of experiment samples related to heat stress are also in one cluster. Although it would be interesting to systematically inspect the common characteristics of the samples in terms of whether the perturbation experiments affect common signaling pathways, such an analysis requires broad and in depth knowledge of yeast genetics, which is beyond the expertise of our group.

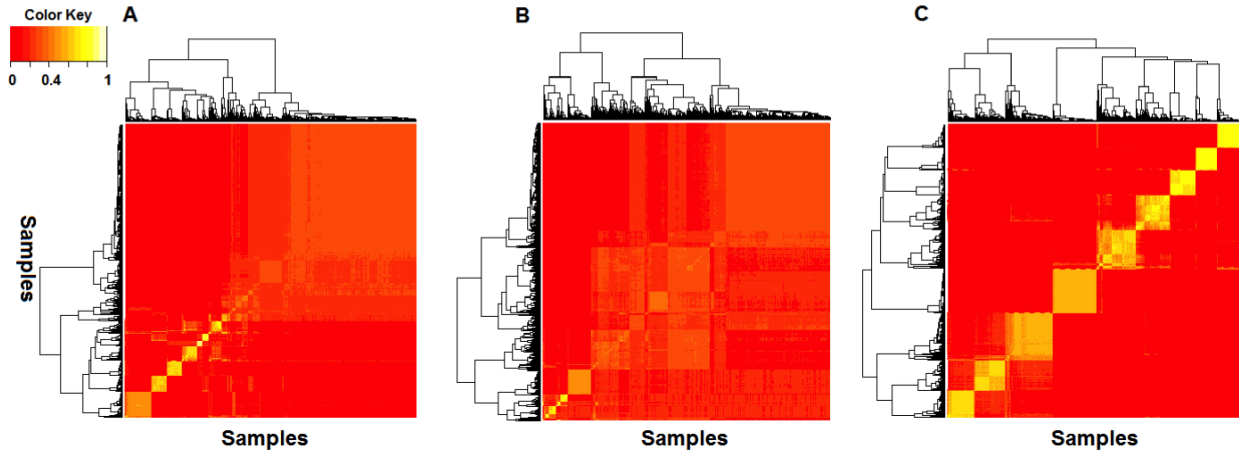


Figure 6.8. Clustering of experiment samples represented using original gene expression data (A), NMF metagenes (B) and expected state of hidden nodes in the first hidden layer (C). Consensus clustering results show how consistently a set of samples is assigned into a common cluster during repeated clustering experiments using samples with replacement from a dataset. A yellow box indicates a set of samples that are consistently assigned to a common cluster, and the brightness of yellow reflects the consistency.

6.3.6 Information embedded in data is consistently represented in different hidden layers

We hypothesized that, in a successful hierarchical representation of a dataset, the information embedded in data should be consistently encoded by different hidden layers, even though two different layers are of different dimensionality and identity of the nodes are totally different. In other words, when a sample is represented by the state of the nodes from different hidden layers, the characteristics that distinguish this sample from others (or make it similar to others) should be retained despite being represented by the nodes from different layers. To test this hypothesis, we first performed consensus clustering using nodes from different hidden layers as features to get sample clusters, and then we compare if members within a cluster derived using one representation significantly overlap with the members from another cluster derived using a

different representation. Figure 6.9 shows the results of assessing the overlaps of the samples clusters derived using the nodes from the 1st and 2nd hidden layers as features respectively. The results indicate that the majority of the clusters derived with different representations agree. Interestingly, a cluster derived from the 2nd hidden layer as features subsumes (maps to) two clusters derived using the 1st hidden layer as features, indicating that the 2nd layer captures more general information. The results indicate that, even though the dimensionality and identity of features of each hidden layer are significantly different, the information encoded by the hidden nodes in a sparse autoencoder is preserved across different layers.

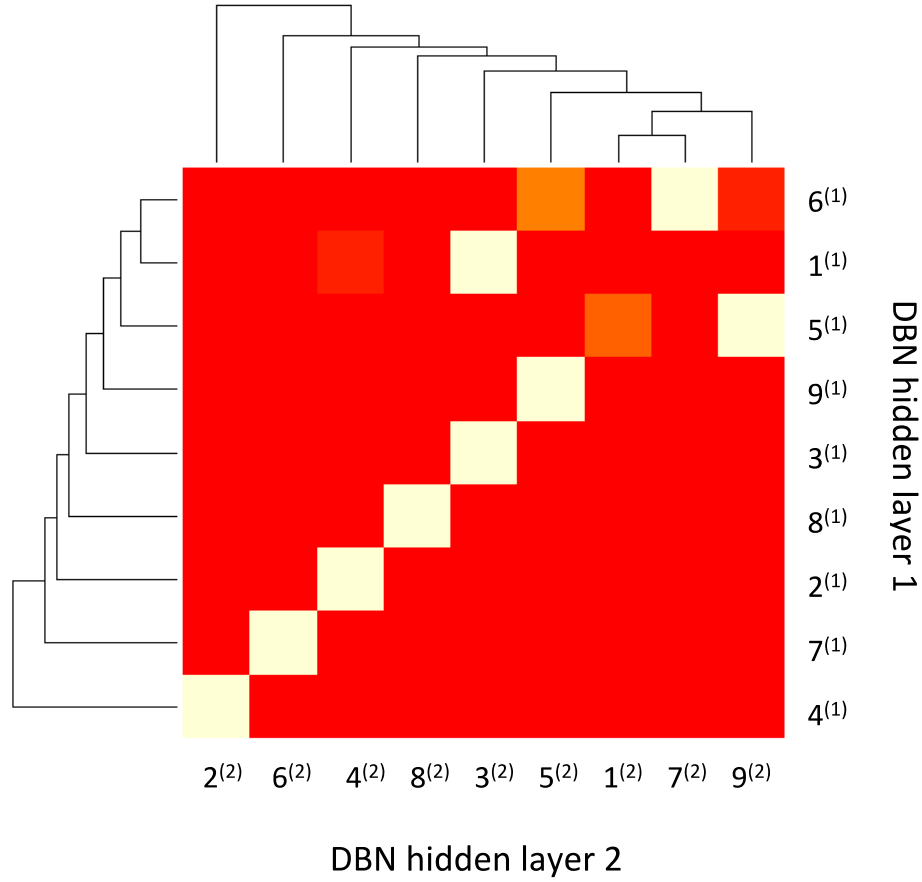


Figure 6.9. Cluster by cluster clustering for clusters in the 1st hidden layer and the 2nd hidden layer.

The x-axis is the 9 sample clusters using the states of hidden units in the 2nd hidden layer indexed by the superscript (2). The y-axis is the 9 samples clusters using the state of hidden units in the 1st hidden layer indexed by the superscript (1). For example, 1⁽¹⁾ is the first sample cluster using the states of hidden units in the 1st hidden layer and 1⁽²⁾ is the first sample cluster using the states of hidden units in the 2nd hidden layer. A yellow box indicates that the members of two clusters (derived from different representation) significantly overlap, with the brightness of yellow reflecting the degree of overlap.

6.3.7 Incorporating prior domain knowledge into DBN model

Incorporating prior domain knowledge into the model (e.g., Bayesian network) was shown to increase the power of prediction/reconstruction (Zou and Conzen 2005). We hypothesized that the reconstruction power of the model could be improved by integrating prior knowledge of TF-gene interactions into the observed input data.

Table 9 lists the reconstruction error of the DBN with and without prior knowledge. A student's t-test was used to test whether the DBN with prior knowledge is significantly different from the one without prior knowledge. The p-value calculated is 0.047, which shows that the two models are significantly different from each other. Table 9 shows that the model with prior knowledge does not reduce the reconstruction error a lot. One of the reasons could be that the resulting reconstruction error is partly decided on the TF coverage of the real data. The information embedded in the observed data does not reflect the regulation of all TFs. Therefore, the prediction is not dramatically improved when the comprehensive prior knowledge of TF-gene interactions is added to the input data.

Table 9. Reconstruction error of DBN with and without prior knowledge

Model	Reconstruction Error (214:100:50:25)
DBN	148.94
DBN with prior knowledge	140.65

6.4 CONCLUSION

In this study, we investigated the utility of contemporary deep hierarchical models to learn a distributed representation of statistical structures embedded in transcriptomic data. We show that such a model is capable of learning biologically sensible representations of the data and revealing novel insights regarding the machinery regulating gene expression. We anticipate that such a model can be used to model more complex systems, such as perturbed signaling systems in cancer cells, thus directly contributing to the understanding of disease mechanisms in translational medicine.

7.0 OVERALL CONCLUSION

This dissertation shows the utility of DLMs, including deep belief network (DBN) and its derived models, in simulating the hierarchical cellular signaling system. It outperformed state-of-the-art non-hierarchical models such as principle component analysis (PCA), non-negative matrix factorization (NMF), support vector machine (SVM) and generalized linear model.

Deep hierarchical models are particularly suited for modeling cellular signaling systems, because signaling molecules in cells are organized as a hierarchical network and information in the system is compositionally encoded. This dissertation indicates that DLMs are capable of capturing the complex information embedded in transcriptomic and proteomic data. This dissertation also demonstrates the feasibility of using DLMs to simulate cellular signaling systems. To our best knowledge, there were no publications using DLMs to study cellular signaling system before, and our previous publication (Chen, Cai et al. 2015) is the first. Our study leads to a new direction of using DLMs to model large “omics” data to gain in-depth knowledge of cellular signaling systems.

8.0 OVERALL DISCUSSION

In this dissertation, I mainly studied transcriptomic gene expression data. However, understanding how the underlying systems in living organisms operate requires the integration of large omics data, such as genomics, transcriptomics, proteomics, epigenomic, metagenomics and metabolomics. Many studies have shown that the prediction of the model would be improved and more comprehensive when more data types are incorporated. The biomedical field is a field where multiple types of data could be assimilated naturally such as gene expression data, mutation data and copy number alteration data. Meanwhile, DLMs, such as multimodal deep belief network, provide a novel approach to simultaneously model multiple types of “omics” data and learn common representations in an “integromics” fashion

Although DLMs have many advantages over other machine learning models such as its hierarchical structure, there are still many challenges to applying DLMs to biomedical fields. 1) **Dataset.** Interestingly, in contrast to the training of DLMs in a machine learning setting such as object recognition in image analysis where usually a large number of training cases is required, this dissertation shows that DLMs performed very well given a moderate size of training cases. This indicates that biological data tend to have strong signals that can be captured by DLMs with relative ease. We believe that the performance of DLMs will be better with more training cases. However, compared with the large training set available for tasks such as image recognition, biomedical data is relatively hard to acquire due to attributes such as privacy restrictions and

ethical requirements. Another problem is data normalization. Biomedical data is often collected under different conditions and platforms. Using the collection of microarray gene expression data as an example, one of the concerns of normalization is the heterogeneity among probe design within microarray platforms, laboratory variations, and methods of data preprocessing (Dozmorov and Wren 2011). It's challenging to choose criteria for normalizing and combining data from various sources. 2) **Representation of hidden units.** As a representation-learning method, deep learning could discover the representation automatically for classification (LeCun 2015). It is easy to evaluate the performance of the model using classification error. However, it's hard to tell what each hidden unit in the structure represents and thus difficult to evaluate the knowledge learnt from the model.

Even though the limitations exist, the future application of DLMs in the biomedical field is still promising. We expect that DLMs will be more widely used in systems biology. DLMs have been shown to successfully learn signaling pathways from yeast transcriptomic data. If DLMs could also model signaling pathways in human, we could apply them to predict drug sensitivity and design personalized cancer therapy. Different from traditional cancer therapy, we could treat patients not only based on their cancer types but also based on the specific signaling pathways perturbed. For example, given a cancer patient's personalized genomic data, we could use DLMs to study how cells encode the signals regulating gene expression, and to detect which signaling pathway is perturbed in a specific pathological condition, e.g., cancer. We could use the knowledge learnt to treat cancer patients with drugs targeting the perturbed signaling pathways.

9.0 FUTURE WORK

9.1.1 Using DLMS to perform translational studies of human cancer

DLMS have been used to study yeast signaling systems, we want to test whether they could be used to represent more complex information embedded in human signaling pathways and utilize the novel features learned by DLMS in personalized cancer medicine. DLMS trained with gene expression data can learn abstract features reflecting cellular signals from training samples, and it can infer the state of signals in test samples. We could use these signal-oriented features learned from DLMS to represent cancers samples and perform the following translational studies: 1) identifying new cancer subtypes; 2) building statistical models for predicting cancer patient survival and 3) predicting drug sensitivity of cell lines in a signaling-pathway-oriented fashion.

9.1.2 Using multimodal DBNs to incorporate more data types to study signaling networks

The multimodal DBN is a promising model for analyzing signaling pathways due to the fact that molecular phenotype readout is not just limited to gene expression. Other data types, such as mutation data and copy number variation data, could also be used as phenotype readout. We could use various types of data together to capture more information about biological components. With the availability of multimodal DBN, the common information among different

data types could be captured by a layer of joint representation above all the high-level representations of individual data types (Figure 2.16).

APPENDIX A

BASIC METHODS FOR DEEP LEARNING

A.1 SUPERVISED AND UNSUPERVISED LEARNING

- **Supervised learning**

- 1) First, a large data set such as images, labeled with its category is used as the input (LeCun 2015).
- 2) Second, train the model using the input and produce an output in the form of a vector of scores (one for each category).
- 3) Then measure the error between the output scores and the desired scores. The machine modifies its internal adjustable parameters to reduce this error. The adjustable parameters are weights. In a typical deep learning model, there may be hundreds of millions of these adjustable weights, and hundreds of millions of labeled examples with which to train the model.
- 4) To properly adjust the weight vector, the learning algorithm computes a gradient vector that, for each weight, indicates by what amount the error would increase or decrease if

the weight were increased by a tiny amount. The weight vector is then adjusted in the opposite direction to the gradient vector.

- 5) In practice, a procedure called stochastic gradient descent (SGD) is used. This consists of showing the input vector for a few examples, computing the outputs and the errors, computing the average gradient for those examples, and adjusting the weights accordingly.
- 6) The process is repeated for many small sets of examples from the training set until the average of the objective function stops decreasing. It is called stochastic because each small set of examples gives a noisy estimate of the average gradient over all examples.
- 7) After training, the performance of the system is measured on a different set of examples called a test set. This is to test the generalization ability of the machine. For the classification, liner classifiers are used. A two-class liner classifier computes a weighted sum of the feature vector components. If the weighted sum is above a threshold, the input is classified as belonging to a particular category.

- **Un-supervised pre-training (especially useful for relatively small dataset)**

The unsupervised learning procedures could create layers of feature detectors without requiring labeled data. The main task of this method is to reconstruct the original input data. The objective in learning each layer of feature detectors was to be able to reconstruct the raw inputs. By ‘pre-training’ several layers of progressively more complex feature detectors using this reconstruction objective, the weights of a deep network could be initialized to sensible values. A final layer of output units could then be added to the top of the network and the whole deep system could be fine-tuned using standard backpropagation (Hinton, Osindero et al. 2006). For

smaller data sets, unsupervised pre-training helps to prevent overfitting, leading to significantly better generalization when the number of labeled examples is small (Erhan, Bengio et al. 2010).

A.2 RESTRICTED BOLTZMANN MACHINE (RBM)

An RBM is an undirected probabilistic graphical model consisting of a layer of stochastic visible binary variables (represented as nodes in the graph) $v \in \{0,1\}^D$ and a layer of stochastic hidden binary variables $h \in \{0,1\}^F$. A RBM is a bipartite graph in which each visible node is connected to every hidden node (Figure 9.1) and vice versa. The statistical structure embedded in the visible variables can be captured by the hidden variables. The RBM model defines the joint distribution of hidden and visible variables using a Boltzmann distribution as follows:

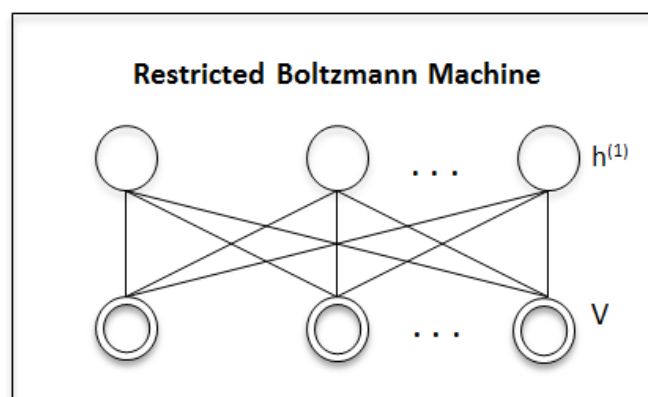


Figure 9.1. Two-layered structure of RBM.

$$Pr(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z(\theta)} \exp(-E(\mathbf{v}, \mathbf{h}; \theta))$$

The energy function E of the state $[\mathbf{v}, \mathbf{h}]$ of the RBM is defined as follows:

$$\begin{aligned} E(\mathbf{v}, \mathbf{h}; \theta) &= -\mathbf{a}^\top \mathbf{v} - \mathbf{b}^\top \mathbf{h} - \mathbf{v}^\top \mathbf{W} \mathbf{h} \\ &= -\sum_{i=1}^D a_i v_i - \sum_{j=1}^F b_j h_j - \sum_{i=1}^D \sum_{j=1}^F v_i h_j w_{ij} \end{aligned}$$

where v_i is the binary state of the visible variable i ; h_j is the binary state of the hidden variable j ; $\theta = \{\mathbf{a}, \mathbf{b}, \mathbf{W}\}$ are the model parameters. a_i represents the bias for the visible variable i and b_j represents the bias for the hidden variable j . w_{ij} represents the weight between the visible variable i and the hidden variable j . The “partition function”, Z , is derived by summing over all possible states of visible and hidden variables.

$$Z(\theta) = \sum_{\mathbf{v}, \mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \theta))$$

The marginal distribution of visible variables is

$$Pr(\mathbf{v}; \theta) = \sum_{\mathbf{h}} Pr(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} \exp(-E(\mathbf{v}, \mathbf{h}; \theta))$$

A.3 LEARNING PARAMETERS OF THE RBM RELATED MODEL

Learning parameters of an RBM model can be achieved by updating the weight matrix and biases using a gradient descent algorithm (delta methods) (Hinton and Salakhutdinov 2006).

$$\mathbf{w}^{t+1} = \mathbf{w}^t + \Delta \mathbf{w},$$

$$\Delta W_{ij} = \epsilon \frac{\partial \log Pr(v)}{\partial w_{ij}} = \epsilon (\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model}),$$

where ϵ is the learning rate; $\langle v_i h_j \rangle_{data}$ is the expected product of the observed data and inferred hidden variables conditioning on observed variables; $\langle v_i h_j \rangle_{model}$ is the expected product of the model-predicted v and h . One approach to derive $\langle v_i h_j \rangle_{model}$ is to obtain samples of \mathbf{v} and \mathbf{h} from a model-defined distribution using Markov chain Monte Carlo (MCMC) methods and then average the product of the samples, which may take a long time to converge. Representing the $\langle v_i h_j \rangle_{model}$ derived MCMC chain after convergence as $\langle v_i h_j \rangle_{\infty}$, one updates the model parameter w_{ij} as follows:

$$\Delta W_{ij} = \epsilon(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{\infty})$$

However, the joint distribution $\langle v_i h_j \rangle_{\infty}$ is hard to calculate. Here, we could use Gibbs sampling to calculate $\langle v_i h_j \rangle_{\infty}$. Gibbs sampling is an algorithm used to generate a sequence of samples from a joint probability distribution. If the conditional distribution of each variable is known, then Gibbs sampling can be applied. To refresh our memory, the derivative of $\log p(v)$ with respect to θ is:

$$\frac{\partial \log p(v)}{\partial \theta} = - \sum_h p(h|v) \frac{\partial E(v, h)}{\partial \theta} + \sum_{\hat{v}, h} p(\hat{v}, h) \frac{\partial E(\hat{v}, h)}{\partial \theta}$$

The first term could be easily calculated from data, because x is the visible data. However, the \hat{x} is all the possible configuration of the predicted input. The complexity grows exponentially as the number of input increases. In this case, we need Gibbs sampling to calculate $E(\hat{v}, h)$ by sampling from $p(\hat{v}, h)$.

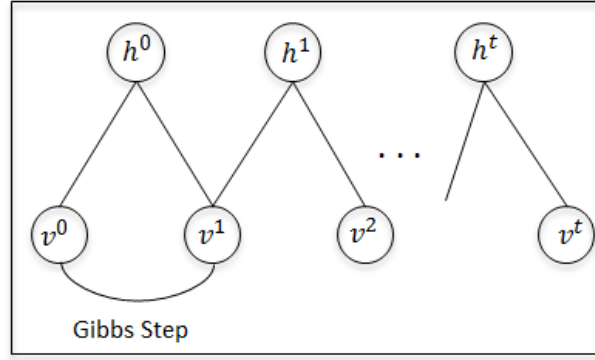


Figure 9.2. Illustration of Gibbs sampling.

Taking the easiest example (Figure 9.2) of the joint probability with two variables v and h in the equation above. We want to sample from $p(v, h)$, but we don't know the exactly joint probability. Here, we use Gibbs sampling to sample repeatedly from two conditional probabilities that are $p(v|h)$ and $p(h|v)$. As the sampling iteration index t tends to infinity, $p(v^t, h^t)$ will converge to $p(v, h)$ (the true joint probability). But the disadvantage is that it's pretty hard to know when equilibrium is reached which is computationally expensive.

The following is the pseudo-code of the process above

$$(v, h)^0 = (v^0, h^0)$$

$$p(h^1) = p(h|v^0)$$

$$p(v^1) = p(v|h^1)$$

$$p(v, h)^1 = p(v^1, h^1)$$

.....

$$p(h^t) = p(h|v^{t-1})$$

$$p(v^t) = p(v|h^{t-1})$$

$$p(v, h)^t = p(v^t, h^t)$$

For RBM, to calculate $\langle v_i h_j \rangle_\infty$, one can alternatively sample the states of hidden variables given visible variables and then sample the states of visible variables given hidden variables (Salakhutdinov, Mnih et al. 2007) based on the following equations:

$$Pr(h_j = 1|v) = \sigma(b_j + \sum_{i=1}^n W_{ij}v_i)$$

$$p_j = Pr(h_j = 1|v)$$

$$Pr(v_i = 1|h) = \sigma(a_i + \sum_{j=1}^m W_{ij}p_j)$$

where $\sigma(x)$ is the logistic function $1/(1 + \exp(-x))$.

As I mentioned above, if the times we sample tends to infinity, the Markov Chain will converge. But the convergence of a MCMC chain may take a long time. To make RBM learning more efficient, we adopted a learning algorithm called contrastive divergence (CD) (Welling and Hinton 2002). Instead of running a MCMC chain for a very large number of steps, CD learning just runs the chain for a small number of steps and minimizes the Kullback-Leibler (KL) divergence between $KL(p_0||p_\infty)$ and $KL(p_n||p_\infty)$ to approximate $\langle v_i h_j \rangle_{model}$ (Carreira-Perpinan and Hinton 2005).

A.3.1 KL Divergence

KL divergence is the difference between two probabilities (P and Q). $KL(P||Q)$ represents the information lost when Q is used to approximate P. P is often the true probability and Q is often the modeled probability. Therefore, a RBM uses KL divergence to measure the difference between the true probability and the approximated probability. KL could be represented as $x \log x$

+ $(1-x)\log(1-x)$ (cross entropy). Some deep learning methods, such as sparse restricted Boltzmann machine, use it to represent the regularization term to restrict the difference between the true activation and the desired activation of a unit.

Therefore, instead of calculating $\Delta W_{ij} = \epsilon(< v_i h_j >_{data} - < v_i h_j >_{\infty})$, the updating algorithm for a parameter of a RBM can be rewritten as follows:

$$\begin{aligned}\Delta W_{ij} &= \epsilon(< v_i h_j >_{data} - < v_i h_j >_{model}) \\ &= \epsilon(< v_i h_j >_{Pr(h|v;w)} - < v_i h_j >_n) \\ \Delta a_i &= \epsilon(< v_i >_{data} - < v_i >_n) \\ \Delta b_j &= \epsilon(< h_j >_{data} - < h_j >_n)\end{aligned}$$

The pseudocode for training a RBM (update of weights and biases) is as follows:

Repeat for t iterations:

- 1) Infer state of hidden units h_{j0} given visible units v_0 $Pr(h_{j0}|v_0)$

$$Pr(h_{j0} = 1|v_0) = \sigma(b_j^t + \sum_{i=1}^n W_{ij}^t v_{i0}) = < h_{j0} >$$

- 2) Gibbs Sampling $< h_{j0} > \rightarrow$ binary matrix h_{j0}

- 3) Infer state of visible units v_{i1} given hidden units h_0 $Pr(v_{i1}|h_0)$

$$Pr(v_{i1} = 1|h_0) = \sigma(a_i^t + \sum_{j=1}^m W_{ij}^t h_{j0}) = < v_{i1} >$$

- 4) Infer state of hidden units h_{j1} given visible units v_1 $Pr(h_{j1}|v_1)$

$$Pr(h_{j1} = 1|v_1) = \sigma(b_j^t + \sum_{i=1}^n W_{ij}^t v_{i1}) = < h_{j1} >$$

- 5) Update parameters (weight between visible i and hidden j , bias of visible and bias of hidden)

$$\begin{aligned}W_{ij}^{t+1} &= W_{ij}^t + \epsilon(< v_{i0}^T h_{j0} > - < v_{i1}^T h_{j1} >) \\ &= W_{ij}^t + \epsilon(v_{i0}^T < h_{j0} > - < v_{i1} >^T < h_{j1} >)\end{aligned}$$

$$a_i^{t+1} = a_i^t + \epsilon(< v_{i0} > - < v_{i1} >)$$

$$b_j^{t+1} = b_j^t + \epsilon(< h_{j0} > - < h_{j1} >)$$

A.3.2 KL Divergence for sparse RBM

Kullback-Leibler divergence (KL divergence) is often used to represent the regularization term (penalty). Let ρ be the desired activation (constant) and $\hat{\rho}_j$ be the average activation of hidden unit j conditional on the input vector (v) across samples. The regularization term is the summation of the KL divergence of all the samples corresponding to a hidden unit.

$$\begin{aligned} \sum_{j=1}^{\text{number of hidden units}} KL(\rho || \hat{\rho}_j) &= \sum_{j=1}^{\text{number of hidden units}} \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j} = \\ &= \sum_{j=1}^{\text{number of hidden units}} -\rho \log \hat{\rho}_j - (1 - \rho) \log(1 - \hat{\rho}_j) + \rho \log \rho + (1 - \rho) \log(1 - \rho) \end{aligned}$$

$$\hat{\rho}_j = \frac{1}{k} \sum_{i=1}^K p(h_j^{(k)} = 1 | v^{(k)})$$

where k is the number of samples.

As illustrated above, we want the activation of the hidden units to be sparse. Therefore, ρ is often set as a small value. We want $\hat{\rho}_j$ to be as close to ρ as possible. $KL(\rho || \hat{\rho}_j)$ is 0 when $\hat{\rho}_j = \rho$. As $\hat{\rho}_j$ diverges from ρ , $KL(\rho || \hat{\rho}_j)$ will increase monotonically.

To train the parameters of the sparse model, we need to maximize the log-likelihood of the data (Ng 2011). To make the activation of the hidden units sparse, we add the regularization term into the optimization problem. As ρ is a constant, we get rid of $\rho \log \rho + (1 - \rho) \log(1 - \rho)$

ρ) in the KL term. The term that is left is $-\rho \log \hat{\rho}_j - (1 - \rho) \log(1 - \hat{\rho}_j)$ which is the cross entropy term mentioned by (V Nair 2009). Thus, given a training set $[v^{(1)}, v^{(k)}, \dots, v^{(K)}]$ comprising K samples, we pose the following optimization problem:

$$\begin{aligned} \arg \min_{\{w, c, b\}} & - \sum_{k=1}^K \log \sum_h Pr(v^{(k)}, h^k) + \lambda \left(\sum_{j=1}^J p_j^{(k)} \log h_j^{(k)+} + (1 - p_j^{(k)}) \log (1 - h_j^{(k)+}) \right) \\ h_j^{(k)+} &= \frac{1}{K} \sum_{k=1}^K P(h_j^{(k)} = 1 | v^{(k)}) \\ p_j^{(k)} &= \rho \end{aligned}$$

The derivative of the regularization term is

$$\begin{aligned} & \lambda \left((-\rho \log \hat{\rho}_j - (1 - \rho) \log(1 - \hat{\rho}_j) + \rho \log \rho + (1 - \rho) \log(1 - \rho)) \right)' \\ &= -\lambda (\rho \log \hat{\rho}_j + (1 - \rho) \log(1 - \hat{\rho}_j))' = -\lambda \left(\frac{\rho}{\hat{\rho}_j} + \frac{1 - \rho}{1 - \hat{\rho}_j} \right) = \lambda \frac{-\rho + \hat{\rho}_j}{\hat{\rho}_j(1 - \hat{\rho}_j)} \\ &= \lambda \left(-\frac{\rho}{\hat{\rho}_j} + \frac{1 - \rho}{1 - \hat{\rho}_j} \right) = \lambda \frac{-\rho + \hat{\rho}_j}{\hat{\rho}_j(1 - \hat{\rho}_j)} \end{aligned}$$

From the inference above, we could see that the derivative of the penalty term is proportional to $\lambda(\hat{\rho}_j - \rho)$. As mentioned in the RBM section (Appendix A.3), the update of parameters for a RBM without the addition of the regularization term is as follows:

$$\Delta w_{ij} = \epsilon (\langle v_i^+ h_j^+ \rangle - \langle v_i^- h_j^- \rangle)$$

$$\Delta b_j = \epsilon (\langle h_j^+ \rangle - \langle h_j^- \rangle)$$

$$\Delta c_i = \epsilon (\langle v_i^+ \rangle - \langle v_i^- \rangle)$$

The form of update changes when the regularization term is added. The regularization term only penalizes the activation of the hidden units. So only the update of the weights and the biases of hidden units is changed; the update of the biases of visible units is still the same.

$$\Delta w_{ij} = \epsilon(\langle v_i^+ h_j^+ \rangle - \langle v_i^- h_j^- \rangle) - \lambda \langle v_i^+ (h_j^+ - \rho_j) \rangle$$

$$\Delta b_j = \epsilon(\langle h_j^+ \rangle - \langle h_j^- \rangle) - \lambda \langle h_j^+ - \rho_j \rangle$$

$$\Delta c_i = \epsilon(\langle v_i^+ \rangle - \langle v_i^- \rangle)$$

A.4 BACKPROPAGATION

A.4.1 Goal of backpropagation

After we finish feed-forward learning, backpropagation is applied to update the weight between each two-neighbored layer to minimize the cost function (output loss, the difference between the true output and the predicted output).

A.4.2 Intuition behind backpropagation

The idea behind backpropagation is as follows: 1) We first use “feedforward” to compute all the nodes’ activations including the activation of the output layer. 2) We calculate the loss function that represents the difference between the predicted output and the true output. 3) We compute the “error term” for each node in each layer that measures how much the node is responsible for the output error. 4) The parameters in each layer are updated based on the partial derivatives of the “error term” on parameters of weight and bias. To reduce the cost function (error term), we repeatedly take steps of gradient descent until convergence or certain criteria are met.

A.4.3 Equation inference and the pseudo-code

The cost function (output loss, the difference between true output and the predicted output) I list below is the square-error cost function that is represented as

$$J(W, b; x, y) = \frac{1}{2} \|y - h_{W,b}(x)\|^2$$

We compute the “error term” $\delta_i^{(l)}$ that measures how much each node i in layer l is responsible for the output loss.

$$\delta_i^{(l)} = \frac{\partial L^{(l)}}{\partial Z_i^{(l)}} = \frac{\partial L^{(l)}}{\partial a_i^{(l)}} \cdot \frac{\partial a_i^{(l)}}{\partial Z_i^{(l)}} = \frac{\partial L^{(l)}}{\partial a_i^{(l)}} \cdot f'(Z_i^{(l)})$$

Assuming the index of the output layer is n_l , then

$$\begin{aligned} \delta_i^{(n_l)} &= \frac{\partial L^{(n_l)}}{\partial Z_i^{(n_l)}} = \frac{\partial L^{(n_l)}}{\partial a_i^{(n_l)}} \cdot \frac{\partial a_i^{(n_l)}}{\partial Z_i^{(n_l)}} = \frac{\partial L^{(n_l)}}{\partial a_i^{(n_l)}} \cdot f'(Z_i^{(n_l)}) \\ &= \frac{\partial}{\partial a_i^{(n_l)}} \frac{1}{2} \|y - h_{W,b}(x)\|^2 \cdot f'(Z_i^{(n_l)}) = -(y_i - a_i^{(n_l)}) \cdot f'(Z_i^{(n_l)}) \end{aligned}$$

For the other layers except the output layer, a weighted average of the error terms of the nodes that uses $a_i^{(l)}$ as an input is

$$\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} w_{ji}^{(l)} \delta_j^{(l+1)} \right) \cdot f'(Z_i^{(l)})$$

The partial derivative of parameters:

$$\begin{aligned} \frac{\partial J(W, b; x, y)}{\partial w_{ij}^{(l)}} &= \frac{\partial J(W, b; x, y)}{\partial Z_i^{(l+1)}} \cdot \frac{\partial Z_i^{(l+1)}}{\partial w_{ij}^{(l)}} = \delta_i^{(l+1)} a_j^{(l)} = a_j^{(l)} \delta_i^{(l+1)} \\ \frac{\partial J(W, b; x, y)}{\partial b_i^{(l)}} &= \frac{\partial J(W, b; x, y)}{\partial Z_i^{(l+1)}} \cdot \frac{\partial Z_i^{(l+1)}}{\partial b_i^{(l)}} = \delta_i^{(l+1)} \cdot 1 = \delta_i^{(l+1)} \end{aligned}$$

Given

$$z_i^{(l+1)} = \sum W_{ij}^{(l)} \cdot a_j^{(l)} + b_i^{(l)}$$

The pseudo-code of backpropagation (from Andrew's note) (Ng 2011)

1. Perform a feedforward pass, computing the activation for layer L_2 , and L_3 , and so on up to the output layer L_{n_l} .

2. For each output unit i in layer n_l (the output layer), set loss function

$$\delta_i^{(n_l)} = \frac{\partial}{\partial z_i^{(n_l)}} \frac{1}{2} \|y - h_{W,b}(x)\|^2 = -(y_i - a_i^{(n_l)}) \cdot f'(z_i^{(n_l)})$$

3. For $l = n_l - 1, n_l - 2, n_l - 3, \dots, 2$

For each node i in layer l , set

$$\delta_i^{(l)} = \left(\sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_j^{(l+1)} \right) \cdot f'(z_i^{(l)})$$

4. Compute the desired partial derivatives, which are given as

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(W, b; x, y) = a_j^{(l)} \delta_i^{(l+1)}$$

$$\frac{\partial}{\partial b_i^{(l)}} J(W, b; x, y) = \delta_i^{(l+1)}$$

APPENDIX B

PSEUDO-CODE

B.1 LEARNING A DBN

The pseudo-code for training a 4-layered DBN is as follows:

Input: Binary data matrix

Output: recognition and generative weights

- 1) Randomly initialize parameters
 - 2) Train RBM for layer 1
 - 3) Train RBM for layer 2
 - 4) Train RBM for layer 3
 - 5) Train RBM for layer 4
 - 6) Backpropagation
-

B.2 VISUALIZATION USING SAMPLING

Pseudo-code for the visualization using sampling:

Assume that the hidden unit i in the layer j that we are interested in is h_{ij} .

Input:

The weights and biases learned by training four-layered sparse DBN

Output:

$E[\mathbf{x}|h_i^j = 1]$ (h_{ij} is the hidden unit i in the hidden layer j)

Procedure

- 1) Randomly generate the data matrix in the layer $h^{(j)}$. We need to damp unit i to always be 1, and the rest to be randomly sampled between 0 and 1. The data matrix generated is denoted as S .

Table 10. Generated states of hidden units in the hidden layer j

$h^{(j)}$	h_1^j	h_i^j	...	h_s^j
Sample1		1		
...		1		
Sample100		1		

- 2) Sample between the top two layers $h^{(j)}$ and $h^{(j-1)}$ until convergence (likes 500 epochs)

For iteration t ,

$$Pr(h^{(j-1)t} | h^{(j)t-1}) = \sigma\left(a + \sum W * h^{(j)t-1}\right) = \langle h^{(j-1)t} \rangle$$

$$h^{(j-1)t} = Pr(h^{(j-1)t} | h^{(j)t-1}) > rand(numcases, numhid)$$

$$Pr(h^{(j)t} | h^{(j-1)t}) = \sigma\left(b + \sum W * h^{(j-1)t}\right) = \langle h^{(j)t} \rangle$$

- 3) Check the distribution of $\langle h^{(j-1)} \rangle$ to make sure that the activation of hidden units is sparse. We let

maximum of 15% of the nodes to be active

- 4) Do the top-down sampling. Sample between every two layers below and stop at the first hidden layer.

- 5) Calculate the probability of the visible units

$$Pr(v | h^{(1)}) = \sigma\left(a + \sum W * h^{(1)}\right) = \langle v \rangle$$

- 6) Calculate $E(v | h^{(1)})$ by averaging $\langle v \rangle$ across samples
-

B.3 MODEL SELECTION

Table 11 and Table 12 list the quantitative comparison between DBNs with different parameter settings from the aspects of reconstruction error and biological representation respectively. We used it to choose an optimal parameter setting for the DBN and sparse DBN.

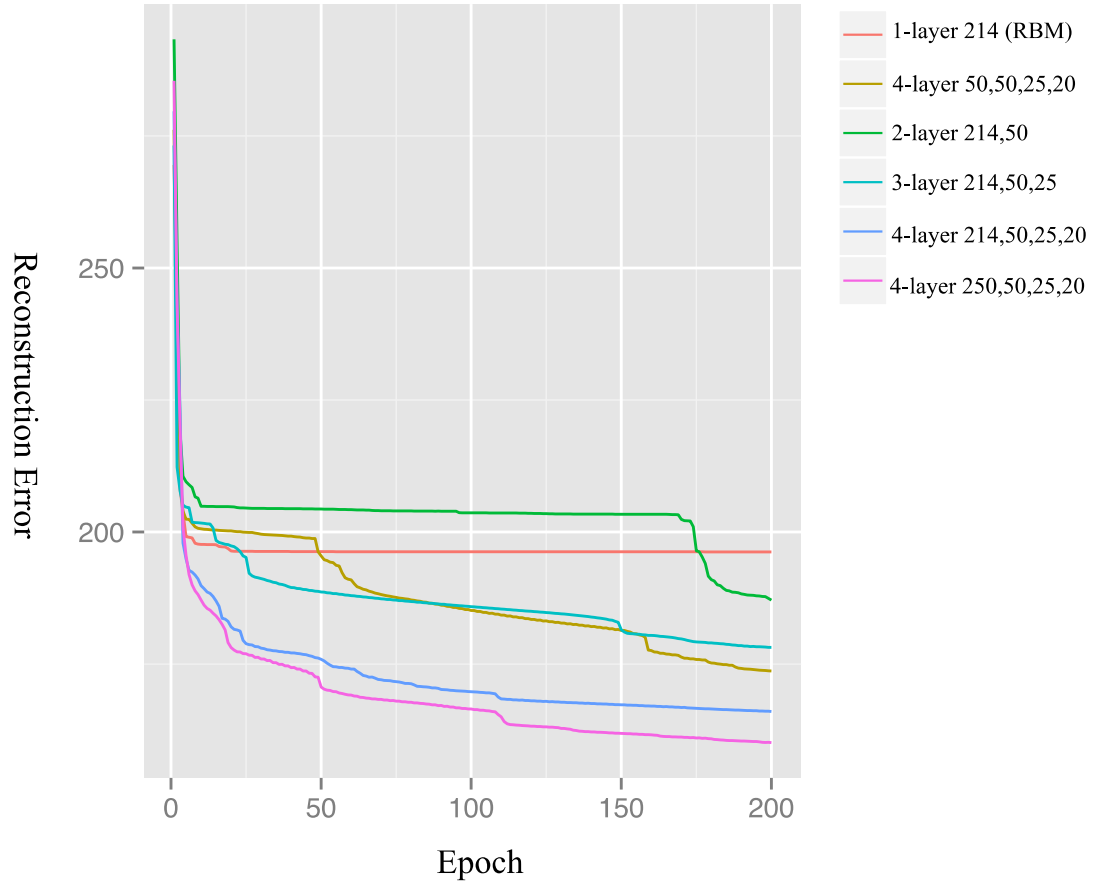


Figure 9.3. Comparison of reconstruction errors corresponding to DBN models with different settings.

The “Reconstruction Error” is the error between the reconstructed gene expression data and the observed gene expression data (Table 11). The number of “Sig-Enrich TF-node Pairs” is the number of values with significant p -values in the mapping matrix. The “Number of Hidden Nodes” is the number of unique hidden units mapped to TFs. The “Number of TFs” is the number of unique TFs mapped to hidden units. The “Average of Enrich p -value” is the average of the negative log of all the significant p -values (Table 12). We used these five criteria together to compare different settings. Table 11 and Table 12 show that the hierarchical four-layered DBN works much better than the simple one-layered DBN (RBM), two-layered and three-

layered DBN (Figure 9.3). Besides, with the increase of hidden units in the first hidden layer, the reconstruction error is reduced and the mapping between hidden units and TFs becomes better. However, the advantage becomes not apparent when the number of hidden units in the first hidden layer exceeds 200. Besides, the reconstruction error increases dramatically when the number of samples used to train the model is reduced.

Table 11. Comparison of reconstruction errors among DBN models with different settings

Parameter Settings	Reconstruction Error
4-layer 50,50,25,20	171.30
4-layer 100,50,25,20	169.12
4-layer 150,50,25,20	166.68
4-layer 214,50,25,20	164.12
4-layer 250,50,25,20	164.03
4-layer 300,50,25,20	163.87
3-layer 214,50,25	177.68
2-layer 214,50	187.12
1-layer 214 (RBM)	196.22
Number of samples: 800	
4-layer 214,50,25,20	178.15
Number of samples: 500	
4-layer 214,50,25,20	187.12
Number of samples: 100	
4-layer 214,50,25,20	200.12

Purple color represents a 4-layer DBN with different number of hidden units in each layer. Orange color represents a 3-layer DBN. Dark blue color represents a 2-layer DBN. Green color represents a RBM. Light blue color represents a 4-layer DBN with different number of training samples.

Table 12. Quantitative comparison among DBN models with different settings

Parameter Settings	Number of Sig-Enrich TF-node Pairs	Number of Hidden Nodes	Number of TFs	Average of Enrich p- value (-log)
4-layer 50,50,25,20	41	24	38	5.594555
4-layer 100,50,25,20	77	43	57	5.640747
4-layer 150,50,25,20	92	51	65	5.546482
4-layer 214,50,25,20	110	59	78	5.582986
4-layer 250,50,25,20	128	76	85	5.653609
4-layer 300,50,25,20	165	91	110	5.658547
3-layer 214,50,25	80	32	40	5.674582
2-layer 214,50	63	20	25	5.621673
1-layer 214 (RBM)	42	19	16	5.619441
Number of samples: 800 4-layer 214,50,25,20	85	45	50	5.770911
Number of samples: 500 4-layer 214,50,25,20	70	29	41	5.580622
Number of samples: 100 4-layer 214,50,25,20	50	25	28	6.140063

Purple color represents a 4-layer DBN with different number of hidden units in each layer. Orange color represents a 3-layer DBN. Dark blue color represents a 2-layer DBN. Green color represents a RBM. Light blue color represents a 4-layer DBN with different number of training samples.

APPENDIX C

Supplementary material for section 4.0 could be found at www.sciencesignaling.org/cgi/content/full/6/299/rs14/DC1.

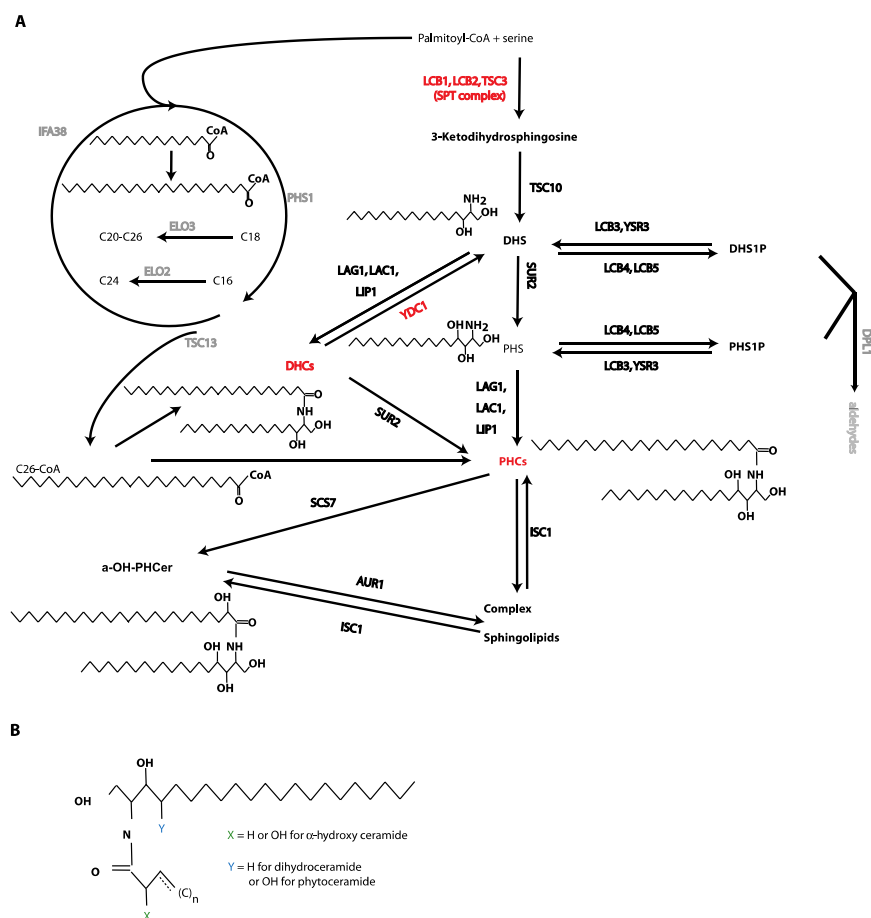


Figure 9.4. *S. cerevisiae* sphingolipid metabolism. (A) Complete sphingolipid metabolic pathway with explicit examples of ceramide structures of each ceramide subspecies investigated. Experimentally manipulated enzymes are highlighted with red, as are the DHCs and PHCs. (B) Generic ceramide structure with a C18 sphingoid

base indicating placement of hydroxyl groups of α -hydroxy and PHC species. A double bond is indicated at the third carbon of the fatty acid, but ceramide species with monounsaturated fatty acids may vary in placement of the double bond.

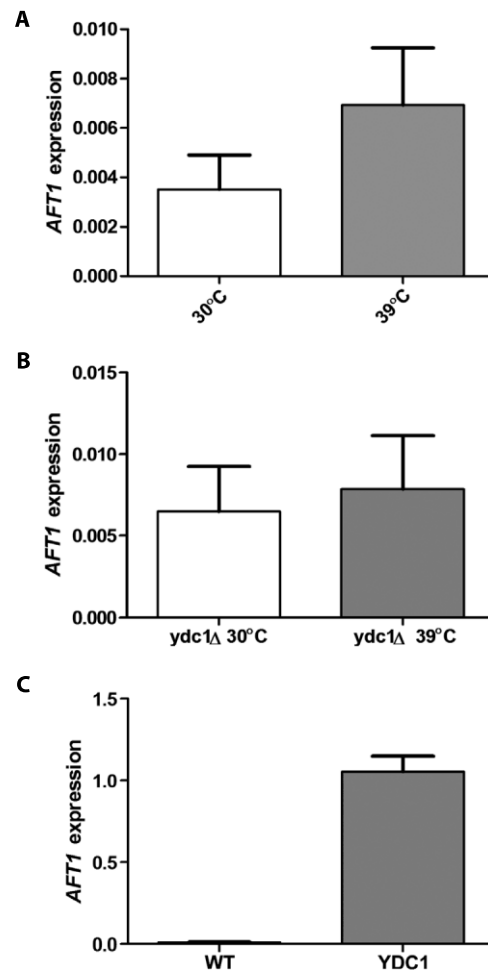


Figure 9.5. Role of Ydc1 in mediating the impact of heat stress on gene expression. (A) Effect of heat stress on AFT1 expression in wild-type (WT) yeast cells (n = 6 and 4, for 30° and 39°C, respectively). (B) Effect of heat stress on AFT1 expression in the ydc1D strain (n = 4, for 30° and 39°C). (C) Effect of overexpression of YDC1 on AFT1 expression at 30°C (n = 6 and 2, for WT and +YDC1, respectively). Data are shown as means \pm SE except for in (C), where the data are shown as average and half of the range for the two +YDC1 measurements.

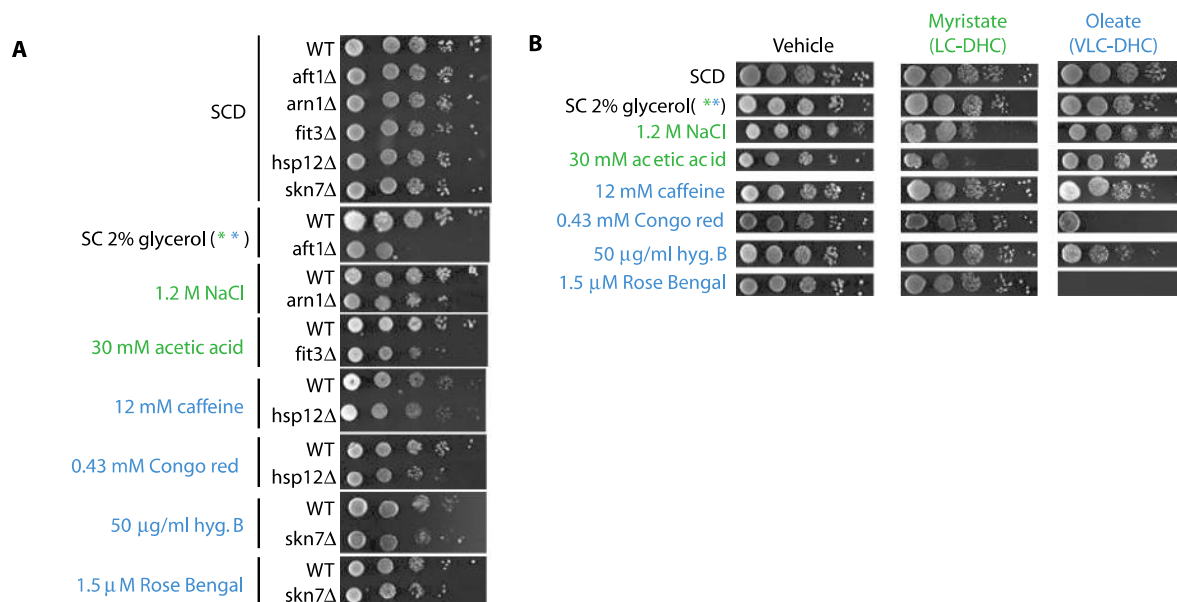


Figure 9.6. Experimental validation of lipid-dependent phenotypes. (A) Confirmation of published genetic phenotypes as positive controls. Published phenotypes for deletion mutants from the LC-DHC-sensitive gene module (green font) or the VLC-DHC-sensitive gene module (blue font) were used to predict ceramide and fatty acid-specific growth defects. One deletion mutant phenotype was confirmed for each treatment used. Conditions were selected from the literature on the basis of phenotypes of genes within each module (Giaever, Chu et al. 2002, Li, Dean et al. 2002, Sambade, Alba et al. 2005, Brombacher, Fischer et al. 2006, Karreman and Lindsey 2007, N. Mira 2010). (B) Validation of LC-DHC- or VLC-DHC-sensitive phenotypes. Rows: specific phenotypes predicted to manifest in response to C14 or C18.1 DHCs, given the indicated treatment condition. Cells were spotted onto agar containing specified treatment plus vehicle (0.1% ethanol), or saturating (1 mM) myristate or oleate. SCD (SC containing 2% dextrose) is no treatment. Spots represent 1:10 serial dilutions of a single mid-log culture. Green font: phenotypes of the LC-DHC-sensitive module predicted to be induced by myristate treatment. Blue font: phenotypes of the VLC-DHC-sensitive module predicted to be induced by oleate treatment, and 2% glycerol is associated with both modules. Images are representative of triplicate experiments.

BIBLIOGRAPHY

- Adler, J. and I. Parmryd (2010). "Quantifying Colocalization by Correlation: The Pearson Correlation Coefficient is Superior to the Mander's Overlap Coefficient." Cytometry Part A **77A**(8): 733-742.
- Al-Feel, W., J. C. DeMar and S. J. Wakil (2003). "A *Saccharomyces cerevisiae* mutant strain defective in acetyl-CoA carboxylase arrests at the G2/M phase of the cell cycle." Proc Natl Acad Sci U S A **100**(6): 3095-3100.
- Alberts, B., A. Jonson, J. Lewis, M. Raff, K. Roberts and P. Walter (2008). Molecular Biology of the Cell. New York, New York, Garland Science, Taylor & Francis Group, LLC.
- Alipanahi, B., A. Delong, M. T. Weirauch and B. J. Frey (2015). "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning." Nat Biotechnol **33**(8): 831-838.
- Alkim, C., L. Benbadis, U. Yilmaz, Z. P. Cakar and J. M. Francois (2013). "Mechanisms other than activation of the iron regulon account for the hyper-resistance to cobalt of a *Saccharomyces cerevisiae* strain obtained by evolutionary engineering." Metallomics **5**(8): 1043-1060.
- Antoniadis, A., S. Lambert-Lacroix and F. Leblanc (2003). "Effective dimension reduction methods for tumor classification using gene expression data." Bioinformatics **19**(5): 563-570.
- Ashburner, M., C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, G. Sherlock and G. O. Consortium (2000). "Gene Ontology: tool for the unification of biology." Nature Genetics **25**(1): 25-29.
- Ausubel, F. M. (2002). "Short Protocols in Molecular Biology: A Compendium of Methods from Current Protocols in Molecular Biology." Wiley, New York.
- Barron, A. R., J.; Yu, B. (1998). "The Minimum Description Length Principle in Coding and Modeling." Information Theory, IEEE Transactions on **44**(6): 2743-2760.
- Ben-Dor, A., R. Shamir and Z. Yakhini (1999). "Clustering gene expression patterns." J Comput Biol **6**(3-4): 281-297.
- Bengio, Y., A. Courville and P. Vincent (2012). "Representation learning: A review and new perspectives." arXiv.org.

- Bishop, C. M. (2006). Pattern recognition and machine learning, Springer Science+Business Media LLC.
- Bloom, G., I. V. Yang, D. Boulware, K. Y. Kwong, D. Coppola, S. Eschrich, J. Quackenbush and T. J. Yeatman (2004). "Multi-platform, multi-site, microarray-based human tumor classification." Am J Pathol **164**(1): 9-16.
- Boone, C., H. Bussey and B. J. Andrews (2007). "Exploring genetic interactions and networks with yeast." Nature Reviews Genetics **8**(6): 437-449.
- Bradley, A. P. (1997). "The use of the area under the roc curve in the evaluation of machine learning algorithms." Pattern Recognition **30**(7): 1145-1159.
- Brand, A. H., and Norbert Perrimon. (1993). "Targeted gene expression as a means of altering cell fates and generating dominant phenotypes." Development **118**(2): 401-415.
- Brombacher, K., B. B. Fischer, K. Rufenacht and R. I. Eggen (2006). "The role of Yap1p and Skn7p-mediated oxidative stress response in the defence of *Saccharomyces cerevisiae* against singlet oxygen." Yeast **23**(10): 741-750.
- Brott, T., H. P. Adams, C. P. Olinger, J. R. Marler, W. G. Barsan, J. Biller, J. Spilker, R. Holleran, R. Eberle, V. Hertzberg, M. Rorick, C. J. Moomaw and M. Walker (1989). "Measurements of Acute Cerebral Infarction - a Clinical Examination Scale." Stroke **20**(7): 864-870.
- Brown, M., and David G. Lowe. (2005). "Unsupervised 3D object recognition and reconstruction in unordered datasets." IEEE Computational Intelligence Magazine.
- Brown, M. P., W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, Jr. and D. Haussler (2000). "Knowledge-based analysis of microarray gene expression data by using support vector machines." Proc Natl Acad Sci U S A **97**(1): 262-267.
- Brown, S. D. (2011). "Disease model discovery and translation. Introduction." Mamm Genome **22**(7-8): 361.
- Brunet, J. P., P. Tamayo, T. R. Golub and J. P. Mesirov (2004). "Metagenes and molecular pattern discovery using matrix factorization." Proceedings of the National Academy of Sciences of the United States of America **101**(12): 4164-4169.
- C. Glymour, G. C. (1999). "Computation, Causation, and Discovery." MIT Press, Cambridge, MA.
- Cameron, D. A., F. A. Middleton, A. Chenn and E. C. Olson (2012). "Hierarchical clustering of gene expression patterns in the Eomes + lineage of excitatory neurons during early neocortical development." BMC Neurosci **13**: 90.

- Carmona-Saez, P., R. D. Pascual-Marqui, F. Tirado, J. M. Carazo and A. Pascual-Montano (2006). "Biclustering of gene expression data by Non-smooth Non-negative Matrix Factorization." BMC Bioinformatics **7**: 78.
- Carreira-Perpinan, M. A. and G. E. Hinton (2005). "On Contrastive Divergence Learning." 33-40.
- Chen, L., C. Cai, V. Chen and X. Lu (2015). "Trans-species learning of cellular signaling systems with bimodal deep belief networks." Bioinformatics.
- Chen, L., C. Cai, V. Chen and X. Lu (2016). "Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model." BMC Bioinformatics **17 Suppl 1**: 9.
- Chen, V. and X. Lu (2013). "Conceptualization of molecular findings by mining gene annotations." BMC Proc **7**(Suppl 7): S2.
- Chen, Y., Y. Li, R. Narayan, A. Subramanian and X. Xie (2016). "Gene expression inference with deep learning." Bioinformatics **32**(12): 1832-1839.
- Clatworthy, J., et al. (2005). "The use and reporting of cluster analysis in health psychology: A review." British journal of health psychology **10**(3): 329-358.
- Collobert, R., and Jason Weston. (2008). "A unified architecture for natural language processing: Deep neural networks with multitask learning." Proceedings of the 25th international conference on Machine learning.
- Cowart, L. A., J. L. Gandy, B. Tholanikunnel and Y. A. Hannun (2010). "Sphingolipids mediate formation of mRNA processing bodies during the heat-stress response of *Saccharomyces cerevisiae*." Biochem J **431**(1): 31-38.
- Cowart, L. A. and Y. A. Hannun (2007). "Selective substrate supply in the regulation of yeast de novo sphingolipid synthesis." J Biol Chem **282**(16): 12330-12340.
- Cowart, L. A., Y. Okamoto, X. Lu and Y. A. Hannun (2006). "Distinct roles for de novo versus hydrolytic pathways of sphingolipid biosynthesis in *Saccharomyces cerevisiae*." Biochem J **393**(Pt 3): 733-740.
- Cowart, L. A., Y. Okamoto, F. R. Pinto, J. L. Gandy, J. S. Almeida and Y. A. Hannun (2003). "Roles for sphingolipid biosynthesis in mediation of specific programs of the heat stress response determined through gene expression profiling." J Biol Chem **278**(32): 30328-30338.
- Cowart, L. A., M. Shotwell, M. L. Worley, A. J. Richards, D. J. Montefusco, Y. A. Hannun and X. Lu (2010). "Revealing a signaling role of phytosphingosine-1-phosphate in yeast." Mol Syst Biol **6**: 349.

- D. J. Klionsky, H. A., et al. (2008). "Guidelines for the use and interpretation of assays for monitoring autophagy in higher eukaryotes. ." Autophagy **4**: 151–175.
- Davis, J. and M. Goadrich (2006). "The relationship between Precision-Recall and ROC curves." Proceedings of the 23rd international conference on Machine learning: 233-240.
- DD Lee, H. S. Algorithms for non-negative matrix factorization. NIPS.
- Detting, M. and P. Buhlmann (2003). "Boosting for tumor classification with gene expression data." Bioinformatics **19**(9): 1061-1069.
- Devarajan, K. (2008). "Nonnegative Matrix Factorization: An Analytical and Interpretive Tool in Computational Biology." Plos Computational Biology **4**(7).
- Dickson, R. C., C. Sumanasekera and R. L. Lester (2006). "Functions and metabolism of sphingolipids in *Saccharomyces cerevisiae*." Prog Lipid Res **45**(6): 447-465.
- Ding, C. and H. Peng (2005). "Minimum redundancy feature selection from microarray gene expression data." J Bioinform Comput Biol **3**(2): 185-205.
- Dohlman, H. G. and J. E. Slessareva (2006). "Pheromone signaling pathways in yeast." Sci STKE **2006**(364): cm6.
- Dolinski, K. and D. Botstein (2005). "Changing perspectives in yeast research nearly a decade after the genome sequence." Genome Research **15**(12): 1611-1619.
- Dombek, P. E., L. K. Johnson, S. T. Zimmerley and M. J. Sadowsky (2000). "Use of repetitive DNA sequences and the PCR to differentiate *Escherichia coli* isolates from human and animal sources." Applied and Environmental Microbiology **66**(6): 2572-2577.
- Dozmorov, M. G. and J. D. Wren (2011). "High-throughput processing and normalization of one-color microarrays for transcriptional meta-analyses." BMC Bioinformatics **12 Suppl 10**: S2.
- Dumitru, E. C., A., Bengio, Y (2010). "Understanding representations learned in deep architectures." Technical report, 1355, Universite de Montreal/DIRO.
- Dutkowski, J., M. Kramer, M. A. Surma, R. Balakrishnan, J. M. Cherry, N. J. Krogan and T. Ideker (2013). "A gene ontology inferred from molecular networks." Nature Biotechnology **31**(1): 38-+.
- Dutkowski, J., K. Ono, M. Kramer, M. Yu, D. Pratt, B. Demchak and T. Ideker (2014). "NeXO Web: the NeXO ontology database and visualization platform." Nucleic Acids Res **42**(Database issue): D1269-1274.
- Eisen, M. B., P. T. Spellman, P. O. Brown and D. Botstein (1998). "Cluster analysis and display of genome-wide expression patterns." Proc Natl Acad Sci U S A **95**(25): 14863-14868.

- Erhan, D., Y. Bengio, A. Courville, P. A. Manzagol, P. Vincent and S. Bengio (2010). "Why Does Unsupervised Pre-training Help Deep Learning?" Journal of Machine Learning Research **11**: 625-660.
- Erhan, D., et al. (2009). "Visualizing higher-layer features of a deep network." Dept. IRO, Université de Montréal, Tech. .
- Esteva, A., Brett Kuprel, and Sebastian Thrun. (2015). "Deep Networks for Early Stage Skin Disease and Skin Cancer Classification." Unpublished manuscript
- Fabrizio, P., F. Pozza, S. D. Pletcher, C. M. Gendron and V. D. Longo (2001). "Regulation of longevity and stress resistance by Sch9 in yeast." Science **292**(5515): 288-290.
- Falcon, S. G., R. (2008). "Hypergeometric testing used for gene set enrichment analysis." Bioconductor case studies. Springer New York: 207-220.
- Friedman, J., T. Hastie and R. Tibshirani (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent." J Stat Softw **33**(1): 1-22.
- Gao, Y., and George Church. (2005). "Improving molecular cancer class discovery through sparse non-negative matrix factorization." Bioinformatics **21**(21): 3970-3975.
- Giaever, G., A. M. Chu, L. Ni, C. Connelly, L. Riles, S. Veronneau, S. Dow, A. Lucau-Danila, K. Anderson, B. Andre, A. P. Arkin, A. Astromoff, M. El-Bakkoury, R. Bangham, R. Benito, S. Brachat, S. Campanaro, M. Curtiss, K. Davis, A. Deutschbauer, K. D. Entian, P. Flaherty, F. Foury, D. J. Garfinkel, M. Gerstein, D. Gotte, U. Guldener, J. H. Hegemann, S. Hempel, Z. Herman, D. F. Jaramillo, D. E. Kelly, S. L. Kelly, P. Kotter, D. LaBonte, D. C. Lamb, N. Lan, H. Liang, H. Liao, L. Liu, C. Luo, M. Lussier, R. Mao, P. Menard, S. L. Ooi, J. L. Revuelta, C. J. Roberts, M. Rose, P. Ross-Macdonald, B. Scherens, G. Schimmack, B. Shafer, D. D. Shoemaker, S. Sookhai-Mahadeo, R. K. Storms, J. N. Strathern, G. Valle, M. Voet, G. Volckaert, C. Y. Wang, T. R. Ward, J. Wilhelmy, E. A. Winzeler, Y. Yang, G. Yen, E. Youngman, K. Yu, H. Bussey, J. D. Boeke, M. Snyder, P. Philippsen, R. W. Davis and M. Johnston (2002). "Functional profiling of the *Saccharomyces cerevisiae* genome." Nature **418**(6896): 387-391.
- Goadrich, M., L. Oliphant and J. Shavlik (2004). "Learning ensembles of first-order clauses for recall-precision curves: A case study in biomedical information extraction." Inductive Logic Programming, Proceedings **3194**: 98-115.
- Goh, H. T., N.; Cord M (2010). "Biasing Restricted Boltzmann Machines to Manipulate Latent Selectivity and Sparsity." NIPS.
- Goldhirsch, A., J. N. Ingle, R. D. Gelber, A. S. Coates, B. Thurlimann, H. J. Senn and m. Panel (2009). "Thresholds for therapies: highlights of the St Gallen International Expert Consensus on the primary therapy of early breast cancer 2009." Ann Oncol **20**(8): 1319-1329.

- Good., P. (1994). "Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses." Springer, Berlin.
- Gruhler, A., J. V. Olsen, S. Mohammed, P. Mortensen, N. J. Faergeman, M. Mann and O. N. Jensen (2005). "Quantitative phosphoproteomics applied to the yeast pheromone signaling pathway." Mol Cell Proteomics **4**(3): 310-327.
- Guenther, G. G., E. R. Peralta, K. R. Rosales, S. Y. Wong, L. J. Siskind and A. L. Edinger (2008). "Ceramide starves cells to death by downregulating nutrient transporter proteins." Proc Natl Acad Sci U S A **105**(45): 17402-17407.
- H Lee, C. E., AY Ng (2008). Sparse deep belief net model for visual area V2. NIPS.
- H Lee, P. P., Y Largman, AY Ng (2009). Unsupervised feature learning for audio classification using convolutional deep belief networks. NIPS.
- Hagan, M. T., et al. (1996). "Neural network design." **20**.
- Hall, M., et al. "The WEKA data mining software: an update." ACM SIGKDD explorations newsletter **11**(1): 10-18.
- Hannun, Y. A. and L. M. Obeid (2008). "Principles of bioactive lipid signalling: lessons from sphingolipids." Nat Rev Mol Cell Biol **9**(2): 139-150.
- Harbison, C. T., D. B. Gordon, T. I. Lee, N. J. Rinaldi, K. D. Macisaac, T. W. Danford, N. M. Hannett, J. B. Tagne, D. B. Reynolds, J. Yoo, E. G. Jennings, J. Zeitlinger, D. K. Pokholok, M. Kellis, P. A. Rolfe, K. T. Takusagawa, E. S. Lander, D. K. Gifford, E. Fraenkel and R. A. Young (2004). "Transcriptional regulatory code of a eukaryotic genome." Nature **431**(7004): 99-104.
- Harris, M. A., J. Clark, A. Ireland, J. Lomax, M. Ashburner, R. Foulger, K. Eilbeck, S. Lewis, B. Marshall, C. Mungall, J. Richter, G. M. Rubin, J. A. Blake, C. Bult, M. Dolan, H. Drabkin, J. T. Eppig, D. P. Hill, L. Ni, M. Ringwald, R. Balakrishnan, J. M. Cherry, K. R. Christie, M. C. Costanzo, S. S. Dwight, S. Engel, D. G. Fisk, J. E. Hirschman, E. L. Hong, R. S. Nash, A. Sethuraman, C. L. Theesfeld, D. Botstein, K. Dolinski, B. Feierbach, T. Berardini, S. Mundodi, S. Y. Rhee, R. Apweiler, D. Barrell, E. Camon, E. Dimmer, V. Lee, R. Chisholm, P. Gaudet, W. Kibbe, R. Kishore, E. M. Schwarz, P. Sternberg, M. Gwinn, L. Hannick, J. Wortman, M. Berriman, V. Wood, N. de la Cruz, P. Tonellato, P. Jaiswal, T. Seigfried, R. White and C. Gene Ontology (2004). "The Gene Ontology (GO) database and informatics resource." Nucleic Acids Res **32**(Database issue): D258-261.
- Hertzberg, R. P. and A. J. Pope (2000). "High-throughput screening: new technology for the 21st century." Curr Opin Chem Biol **4**(4): 445-451.
- Hill, C. S., R. Marais, S. John, J. Wynne, S. Dalton and R. Treisman (1993). "Functional analysis of a growth factor-responsive transcription factor complex." Cell **73**(2): 395-406.

- Hinton G., D. L., et al. (2012). "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." Signal Processing Magazine, IEEE.
- Hinton, G. E. (1992). "How neural networks learn from experience." Scientific American **267**(3): 145-151.
- Hinton, G. E., et al. (2012). "Improving neural networks by preventing co-adaptation of feature detectors." arXiv preprint arXiv:1207.0580
- Hinton, G. E., S. Osindero and Y. W. Teh (2006). "A fast learning algorithm for deep belief nets." Neural Comput **18**(7): 1527-1554.
- Hinton, G. E. and R. R. Salakhutdinov (2006). "Reducing the dimensionality of data with neural networks." Science **313**(5786): 504-507.
- Holter, N. S., M. Mitra, A. Maritan, M. Cieplak, J. R. Banavar and N. V. Fedoroff (2000). "Fundamental patterns underlying gene expression profiles: simplicity from complexity." Proc Natl Acad Sci U S A **97**(15): 8409-8414.
- Huang, S. S. and E. Fraenkel (2009). "Integrating proteomic, transcriptional, and interactome data reveals hidden components of signaling and regulatory networks." Sci Signal **2**(81): ra40.
- Hughes, T. R., M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraborty, J. Simon, M. Bard and S. H. Friend (2000). "Functional discovery via a compendium of expression profiles." Cell **102**(1): 109-126.
- Hughes, T. R., M. D. Robinson, N. Mitsakakis and M. Johnston (2004). "The promise of functional genomics: completing the encyclopedia of a cell." Current Opinion in Microbiology **7**(5): 546-554.
- Jain, A. K., Jianchang Mao, and K. Moidin Mohiuddin. (1996). ""Artificial neural networks: A tutorial." IEEE computer 29.3 (1996): 31-44." IEEE computer **29**(3): 31-44.
- Jin, B., B. Muller, C. Zhai and X. Lu (2008). "Multi-label literature classification based on the Gene Ontology graph." BMC Bioinformatics **9**: 525.
- Jolliffe, I. (2002). "Principal component analysis. ." John Wiley & Sons, Ltd.
- Kanehisa, M. and S. Goto (2000). "KEGG: kyoto encyclopedia of genes and genomes." Nucleic Acids Res **28**(1): 27-30.
- Karatzoglou, A., A. Smola, K. Hornik and A. Zeileis (2004). "kernlab -- An S4 package for kernel methods in R." Journal of Statistical Software **11**(9): 1.

- Karreman, R. J. and G. G. Lindsey (2007). "Modulation of Congo-red-induced aberrations in the yeast *Saccharomyces cerevisiae* by the general stress response protein Hsp12p." Can J Microbiol **53**(11): 1203-1210.
- Khatrri, P. and S. Draghici (2005). "Ontological analysis of gene expression data: current tools, limitations, and open problems." Bioinformatics **21**(18): 3587-3595.
- Kim, P. M. and B. Tidor (2003). "Subsystem identification through dimensionality reduction of large-scale gene expression data." Genome Research **13**(7): 1706-1718.
- Kitagaki, H., L. A. Cowart, N. Matmati, D. Montefusco, J. Gandy, S. V. de Avalos, S. A. Novgorodov, J. Zheng, L. M. Obeid and Y. A. Hannun (2009). "ISC1-dependent metabolic adaptation reveals an indispensable role for mitochondria in induction of nuclear genes during the diauxic shift in *Saccharomyces cerevisiae*." J Biol Chem **284**(16): 10818-10830.
- Kohavi, R., and George H. John. (1997). "Wrappers for feature subset selection." Artificial intelligence **97**(1): 273-324.
- Kolter, T. (2011). "A view on sphingolipids and disease." Chem Phys Lipids **164**(6): 590-606.
- Kong, W., C. R. Vanderburg, H. Gunshin, J. T. Rogers and X. Huang (2008). "A review of independent component analysis application to microarray gene expression data." Biotechniques **45**(5): 501-520.
- Krizhevsky, A., Ilya Sutskever, and Geoffrey E. Hinton. (2012). "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems.
- Kursa, M. B., and Witold R. Rudnicki. (2010). "Feature selection with the Boruta package.".
- Le, Q. (2013). Building high-level features using large scale unsupervised learning. Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference.
- LeCun, Y., and Yoshua Bengio. (1995). "Convolutional networks for images, speech, and time series." The handbook of brain theory and neural networks **3361.10** (10).
- LeCun, Y., et al. (1998). "Gradient-based learning applied to document recognition." Proceedings of the IEEE **86**(11): 2278-2324.
- LeCun, Y., Yoshua Bengio, and Geoffrey Hinton. (2015). "Deep learning." Nature **521**(7553): 436-444.
- Lee, D. D., and H. Sebastian Seung. (2001). "Algorithms for non-negative matrix factorization." Advances in neural information processing systems.

- Lee, H., et al. (2009). "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations." Proceedings of the 26th annual international conference on machine learning.
- Lee, H. E., C.; Ng, A.Y. (2008). "Sparse deep belief net model for visual area V2." Advances in Neural Information Processing Systems.
- Lee, J. T. and J. A. McCubrey (2002). "The Raf/MEK/ERK signal transduction cascade as a target for chemotherapeutic intervention in leukemia." Leukemia **16**(4): 486-507.
- Leung, M. K., H. Y. Xiong, L. J. Lee and B. J. Frey (2014). "Deep learning of the tissue-regulated splicing code." Bioinformatics **30**(12): i121-129.
- Li, S., S. Dean, Z. Li, J. Horecka, R. J. Deschenes and J. S. Fassler (2002). "The eukaryotic two-component histidine kinase Sln1p regulates OCH1 via the transcription factor, Skn7p." Mol Biol Cell **13**(2): 412-424.
- Li, Z., F. J. Vizeacoumar, S. Bahr, J. Li, J. Warringer, F. S. Vizeacoumar, R. Min, B. Vandersluis, J. Bellay, M. Devit, J. A. Fleming, A. Stephens, J. Haase, Z. Y. Lin, A. Baryshnikova, H. Lu, Z. Yan, K. Jin, S. Barker, A. Datti, G. Giaever, C. Nislow, C. Bulawa, C. L. Myers, M. Costanzo, A. C. Gingras, Z. Zhang, A. Blomberg, K. Bloom, B. Andrews and C. Boone (2011). "Systematic exploration of essential yeast gene function with temperature-sensitive mutants." Nat Biotechnol **29**(4): 361-367.
- Liang, M., Z. Li, T. Chen and J. Zeng (2015). "Integrative Data Analysis of Multi-Platform Cancer Data with a Multimodal Deep Learning Approach." IEEE/ACM Trans Comput Biol Bioinform **12**(4): 928-937.
- Liao, J. C., R. Boscolo, Y. L. Yang, L. M. Tran, C. Sabatti and V. P. Roychowdhury (2003). "Network component analysis: Reconstruction of regulatory signals in biological systems." Proceedings of the National Academy of Sciences of the United States of America **100**(26): 15522-15527.
- Liaw, A., and Matthew Wiener. (2002). "Classification and regression by randomForest." R news **2.3**: 18-22.
- Liebermeister, W. (2002). "Linear modes of gene expression determined by independent component analysis." Bioinformatics **18**(1): 51-60.
- Liu, M., C. Huang, S. R. Polu, R. Schneider and A. Chang (2012). "Regulation of sphingolipid synthesis through Orm1 and Orm2 in yeast." J Cell Sci **125**(Pt 10): 2428-2435.
- Lu, S., B. Jin, L. A. Cowart and X. Lu (2013). "From data towards knowledge: revealing the architecture of signaling systems by unifying knowledge mining and data mining of systematic perturbation data." PLoS One **8**(4): e61134.
- Lu, S. and X. Lu (2012). "Integrating genome and functional genomics data to reveal perturbed signaling pathways in ovarian cancers." AMIA Jt Summits Transl Sci Proc **2012**: 72-78.

- Lu, X., M. Hauskrecht and R. S. Day (2004). "Modeling cellular processes with variational Bayesian cooperative vector quantizer." Pac Symp Biocomput: 533-544.
- Lussier, Y. A. and J. L. Chen (2011). "The emergence of genome-based drug repositioning." Sci Transl Med **3**(96): 96ps35.
- M. A. Collart, S. O. (1993). "Preparation of yeast RNA, in Current Protocols in Molecular Biology." John Wiley & Sons Inc., New York: 13.12.11–13.12.15.
- Ma, S. and M. R. Kosorok (2009). "Identification of differential gene pathways with principal component analysis." Bioinformatics **25**(7): 882-889.
- Mao, C., R. Xu, A. Bielawska and L. M. Obeid (2000). "Cloning of an alkaline ceramidase from *Saccharomyces cerevisiae*. An enzyme with reverse (CoA-independent) ceramide synthase activity." J Biol Chem **275**(10): 6876-6884.
- Matmati, N., H. Kitagaki, D. Montefusco, B. K. Mohanty and Y. A. Hannun (2009). "Hydroxyurea sensitivity reveals a role for ISC1 in the regulation of G2/M." J Biol Chem **284**(13): 8241-8246.
- McGonigle, P. and B. Ruggeri (2014). "Animal models of human disease: challenges in enabling translation." Biochem Pharmacol **87**(1): 162-171.
- Min, S., Byunghan Lee, and Sungroh Yoon. (2016). "Deep Learning in Bioinformatics." arXiv preprint
- Mohamed, A. R., G. E. Dahl and G. Hinton (2012). "Acoustic Modeling Using Deep Belief Networks." Ieee Transactions on Audio Speech and Language Processing **20**(1): 14-22.
- Montefusco, D. J., L. J. Chen, N. Matmati, S. J. Lu, B. Newcomb, G. F. Cooper, Y. A. Hannun and X. H. Lu (2013). "Distinct Signaling Roles of Ceramide Species in Yeast Revealed Through Systematic Perturbation and Systems Biology Analyses." Science Signaling **6**(299).
- Montefusco, D. J., B. Newcomb, J. L. Gandy, S. E. Brice, N. Matmati, L. A. Cowart and Y. A. Hannun (2012). "Sphingoid bases and the serine catabolic enzyme CHA1 define a novel feedforward/feedback mechanism in the response to serine availability." J Biol Chem **287**(12): 9280-9289.
- Monti, S. T., P.; Mesirov, J.; Golub, T. (2003). "Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data." Machine Learning **52**(1-2): 91-118.
- Motshwene, P., R. Karreman, G. Kgari, W. Brandt and G. Lindsey (2004). "LEA (late embryonic abundant)-like protein Hsp 12 (heat-shock protein 12) is present in the cell wall and enhances the barotolerance of the yeast *Saccharomyces cerevisiae*." Biochem J **377**(Pt 3): 769-774.

- Mousley, C. J., K. Tyeryar, K. E. Ile, G. Schaaf, R. L. Brost, C. Boone, X. Guan, M. R. Wenk and V. A. Bankaitis (2008). "Trans-Golgi network and endosome dynamics connect ceramide homeostasis with regulation of the unfolded protein response and TOR signaling in yeast." Mol Biol Cell **19**(11): 4785-4803.
- Mullen, T. D., S. Spassieva, R. W. Jenkins, K. Kitatani, J. Bielawski, Y. A. Hannun and L. M. Obeid (2011). "Selective knockdown of ceramide synthases reveals complex interregulation of sphingolipid metabolism." J Lipid Res **52**(1): 68-77.
- Muller, B., A. J. Richards, B. Jin and X. Lu (2009). "GOGrapher: A Python library for GO graph representation and analysis." BMC Res Notes **2**: 122.
- N. Mira, M. T., I. Sá-Correia. (2010). "Adaptative response and tolerance to weak acid stress in *Saccharomyces cerevisiae*: A genome-wide view. ." OMICS **14**: 525–540
- Nair, V., and Geoffrey E. Hinton. (2010). "Rectified linear units improve restricted boltzmann machines." Proceedings of the 27th International Conference on Machine Learning (ICML-10).
- Ng, A. (2011). "Sparse autoencoder." CS294A Lecture notes: 72.
- Ngiam, J., et al. (2011). "Multimodal deep learning." Proceedings of the 28th International Conference on Machine Learning (ICML-11).
- Nguyen, D. V., and David M. Rocke. (2002). " Tumor classification by partial least squares using microarray gene expression data." Bioinformatics **18**(1): 39-50.
- Omar, B. A., J. Vikman, M. S. Winzell, U. Voss, E. Ekblad, J. E. Foley and B. Ahren (2013). "Enhanced beta cell function and anti-inflammatory effect after chronic treatment with the dipeptidyl peptidase-4 inhibitor vildagliptin in an advanced-aged diet-induced obesity mouse model." Diabetologia **56**(8): 1752-1760.
- Payan, A., and Giovanni Montana (2015). "Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks." arXiv preprint
- Peemen, M. (2015). "Deep Learning in a Nutshell: Core Concepts."
- Petersen, T. N., S. Brunak, G. von Heijne and H. Nielsen (2011). "SignalP 4.0: discriminating signal peptides from transmembrane regions." Nature Methods **8**(10): 785-786.
- Petschnigg, J., H. Wolinski, D. Kolb, G. Zellnig, C. F. Kurat, K. Natter and S. D. Kohlwein (2009). "Good fat, essential cellular requirements for triacylglycerol synthesis to maintain membrane homeostasis in yeast." J Biol Chem **284**(45): 30981-30993.
- Plis, S. M., D. R. Hjelm, R. Salakhutdinov, E. A. Allen, H. J. Bockholt, J. D. Long, H. J. Johnson, J. S. Paulsen, J. A. Turner and V. D. Calhoun (2014). "Deep learning for neuroimaging: a validation study." Front Neurosci **8**: 229.

- Posada, D. and T. R. Buckley (2004). "Model selection and model averaging in phylogenetics: Advantages of akaike information criterion and Bayesian approaches over likelihood ratio tests." Systematic Biology **53**(5): 793-808.
- Poussin, C., et al. (2014). "The species translation challenge—A systems biology perspective on human and rat bronchial epithelial cells." Scientific Data **1**.
- Qi, Q., Y. Zhao, M. Li and R. Simon (2009). "Non-negative matrix factorization of gene expression profiles: a plug-in for BRB-ArrayTools." Bioinformatics **25**(4): 545-547.
- QIAGEN. "Mating-Pheromone Response Pathway in Budding Yeast." from [http://www.qiagen.com/us/products/genes and pathways/pathway details.aspx?pwid=283](http://www.qiagen.com/us/products/genes_and_pathways/pathway_details.aspx?pwid=283).
- Raychaudhuri, S., J. M. Stuart and R. B. Altman (2000). "Principal components analysis to summarize microarray experiments: application to sporulation time series." Pac Symp Biocomput: 455-466.
- Reshef, D. N., Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher and P. C. Sabeti (2011). "Detecting novel associations in large data sets." Science **334**(6062): 1518-1524.
- Rhrissorakrai, K., V. Belcastro, E. Bilal, R. Norel, C. Poussin, C. Mathis, R. H. J. Dulize, N. V. Ivanov, L. Alexopoulos, J. J. Rice, M. C. Peitsch, G. Stolovitzky, P. Meyer and J. Hoeng (2015). "Understanding the limits of animal models as predictors of human biology: lessons learned from the sbv IMPROVER Species Translation Challenge." Bioinformatics **31**(4): 471-483.
- Ringnér, M. (2008). "What is principal component analysis?" Nature biotechnology **26**(3): 303-304.
- Ronan C., W. J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. ACM.
- Rumelhart, D. E., Geoffrey E. Hinton, and Ronald J. Williams. (1988). "Learning representations by back-propagating errors." Cognitive modeling **5.3** **1**.
- Saeys, Y., I. Inza and P. Larranaga (2007). "A review of feature selection techniques in bioinformatics." Bioinformatics **23**(19): 2507-2517.
- Salakhutdinov, R., and Geoffrey E. Hinton. (2009). "Deep Boltzmann Machines." AISTATS **1**.
- Salakhutdinov, R., A. Mnih and G. E. Hinton (2007). "Restricted Boltzmann Machines for Collaborative Filtering." Proceedings of the 24th international conference on Machine learning 791-798.
- Sambade, M., M. Alba, A. M. Smardon, R. W. West and P. M. Kane (2005). "A genomic screen for yeast vacuolar membrane ATPase mutants." Genetics **170**(4): 1539-1551.

- SBV IMPROVER. (2013). "SBV IMPROVER: Species Translation Challenge Overview." from <https://www.sbvimprover.com/challenge-2/overview>.
- Shaffer, A. T. (2011). "Emerging Biomarker Science Presents Practical Questions ".
- Simpson, T. I., J. D. Armstrong and A. P. Jarman (2010). "Merged consensus clustering to assess and improve class discovery with microarray data." BMC Bioinformatics **11**: 590.
- Slamon, D. J., B. Leyland-Jones, S. Shak, H. Fuchs, V. Paton, A. Bajamonde, T. Fleming, W. Eiermann, J. Wolter, M. Pegram, J. Baselga and L. Norton (2001). "Use of chemotherapy plus a monoclonal antibody against HER2 for metastatic breast cancer that overexpresses HER2." N Engl J Med **344**(11): 783-792.
- Smith, L. I. (2002). "A tutorial on principal components analysis." **52**.
- Somol, P., and Pavel Pudil. (2001). " Feature selection toolbox." Pattern Recognition **35**(12): 2749-2759.
- Spassieva, S. D., M. Rahmaniyan, J. Bielawski, C. J. Clarke, J. M. Kravaka and L. M. Obeid (2012). "Cell density-dependent reduction of dihydroceramide desaturase activity in neuroblastoma cells." J Lipid Res **53**(5): 918-928.
- Spellman, P. T., G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein and B. Futcher (1998). "Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization." Mol Biol Cell **9**(12): 3273-3297.
- Srivastava, N. (2013). "Improving neural networks with dropout." University of Toronto.
- Srivastava, N., et al. (2014). "Dropout: a simple way to prevent neural networks from overfitting." Journal of Machine Learning Research **15**(1): 1929-1958.
- Srivastava, N. and R. Salakhutdinov (2012). "Multimodal Learning with Deep Boltzmann Machines." NIPS: 2231-2239.
- Storey, J. (2003). "The positive false discovery rate: A Bayesian interpretation and the q-value." Annals of Statistics **31**: 2013–2035.
- Sturn, A., J. Quackenbush and Z. Trajanoski (2002). "Genesis: cluster analysis of microarray data." Bioinformatics **18**(1): 207-208.
- Sturtevant, A. H. (1956). "A Highly Specific Complementary Lethal System in *Drosophila-Melanogaster*." Genetics **41**(1): 118-123.
- Subramanian, A., H. Kuehn, J. Gould, P. Tamayo and J. P. Mesirov (2007). "GSEA-P: a desktop application for Gene Set Enrichment Analysis." Bioinformatics **23**(23): 3251-3253.

- Subramanian, A., P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander and J. P. Mesirov (2005). "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles." Proc Natl Acad Sci U S A **102**(43): 15545-15550.
- Sudhher, P., and V. Kesav Kumar. "Clustering Algorithm Based On Correlation Preserving Indexing."
- Suk, H. I. and D. Shen (2013). "Deep learning-based feature representation for AD/MCI classification." Med Image Comput Comput Assist Interv **16**(Pt 2): 583-590.
- Suykens, J. A., and Joos Vandewalle. (1999). "Least squares support vector machine classifiers." Neural processing letters **9**(3): 293-300.
- Tamanoi, F. (2011). "Ras signaling in yeast." Genes Cancer **2**(3): 210-215.
- Tamayo, P., D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander and T. R. Golub (1999). "Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation." Proc Natl Acad Sci U S A **96**(6): 2907-2912.
- Taylor, G. W. and G. E. Hinton (2009). "Factored conditional restricted Boltzmann Machines for modeling motion style." Proceedings of the 26th Annual International Conference on Machine Learning 1025-1032.
- Tibshirani, R. (1996). "Regression shrinkage and selection via the lasso." Journal of the Royal Statistical Society.: 267-288.
- Toke, D. A. and C. E. Martin (1996). "Isolation and characterization of a gene affecting fatty acid elongation in *Saccharomyces cerevisiae*." J Biol Chem **271**(31): 18413-18422.
- Tong, A. H., M. Evangelista, A. B. Parsons, H. Xu, G. D. Bader, N. Page, M. Robinson, S. Raghizadeh, C. W. Hogue, H. Bussey, B. Andrews, M. Tyers and C. Boone (2001). "Systematic genetic analysis with ordered arrays of yeast deletion mutants." Science **294**(5550): 2364-2368.
- Tsoumakas, G. and I. Katakis (2007). "Multi-Label Classification: An Overview." Data Warehousing and Mining **3**(3): 1-13.
- V Nair, G. H. (2009). 3D object recognition with deep belief nets. NIPS.
- Van Brocklyn, J. R. (2007). "Sphingolipid signaling pathways as potential therapeutic targets in gliomas." Mini-Reviews in Medicinal Chemistry **7**(10): 984-990.
- Vincent, P. L., H.; Bengio, Y. (2008). "Extracting and composing robust features with denoising autoencoders." Proceedings of the 25th international conference on Machine learning.

- Wang, Y., and Fillia Makedon. (2004). "Application of Relief-F feature filtering algorithm to selecting informative genes for cancer classification using microarray data." IEEE Computer Society: 497–498.
- Welling, M. and G. E. Hinton (2002). "A new learning algorithm for Mean Field Boltzmann Machines." Artificial Neural Networks - Icann 2002 **2415**: 351-357.
- Wells, G. B., R. C. Dickson and R. L. Lester (1998). "Heat-induced elevation of ceramide in *Saccharomyces cerevisiae* via de novo synthesis." J Biol Chem **273**(13): 7235-7243.
- Y. A. Hannun, L. M. (2011). "Obeid, Many ceramides. ." J. Biol. Chem **286**: 27855–27862.
- Y., B. (2009). "Learning deep architectures for AI." Foundations and trends® in Machine Learning 2.1: 1-127.
- Yarden, Y. (2001). "The EGFR family and its ligands in human cancer: signalling mechanisms and therapeutic opportunities." European Journal of Cancer **37**: S3-S8.
- Yeger-Lotem, E., L. Riva, L. J. Su, A. D. Gitler, A. G. Cashikar, O. D. King, P. K. Auluck, M. L. Geddie, J. S. Valastyan, D. R. Karger, S. Lindquist and E. Fraenkel (2009). "Bridging high-throughput genetic and transcriptional data reveals cellular responses to alpha-synuclein toxicity." Nat Genet **41**(3): 316-323.
- Zhang, J. H., S. H. Zhang, Y. Wang and X. S. Zhang (2013). "Identification of mutated core cancer modules by integrating somatic mutation, copy number variation, and gene expression data." Bmc Systems Biology **7**.
- Zhao, J., S. Gupta, M. Seielstad, J. Liu and A. Thalamuthu (2011). "Pathway-based analysis using reduced gene subsets in genome-wide association studies." BMC Bioinformatics **12**.
- Zou, H., and Trevor Hastie (2005). "Regularization and variable selection via the elastic net." Journal of the Royal Statistical Society: Series B (Statistical Methodology) **67**(2): 301-320.
- Zou, M. and S. D. Conzen (2005). "A new dynamic Bayesian network (DBN) approach for identifying gene regulatory networks from time course microarray data." Bioinformatics **21**(1): 71-79.